

AD-A074 321

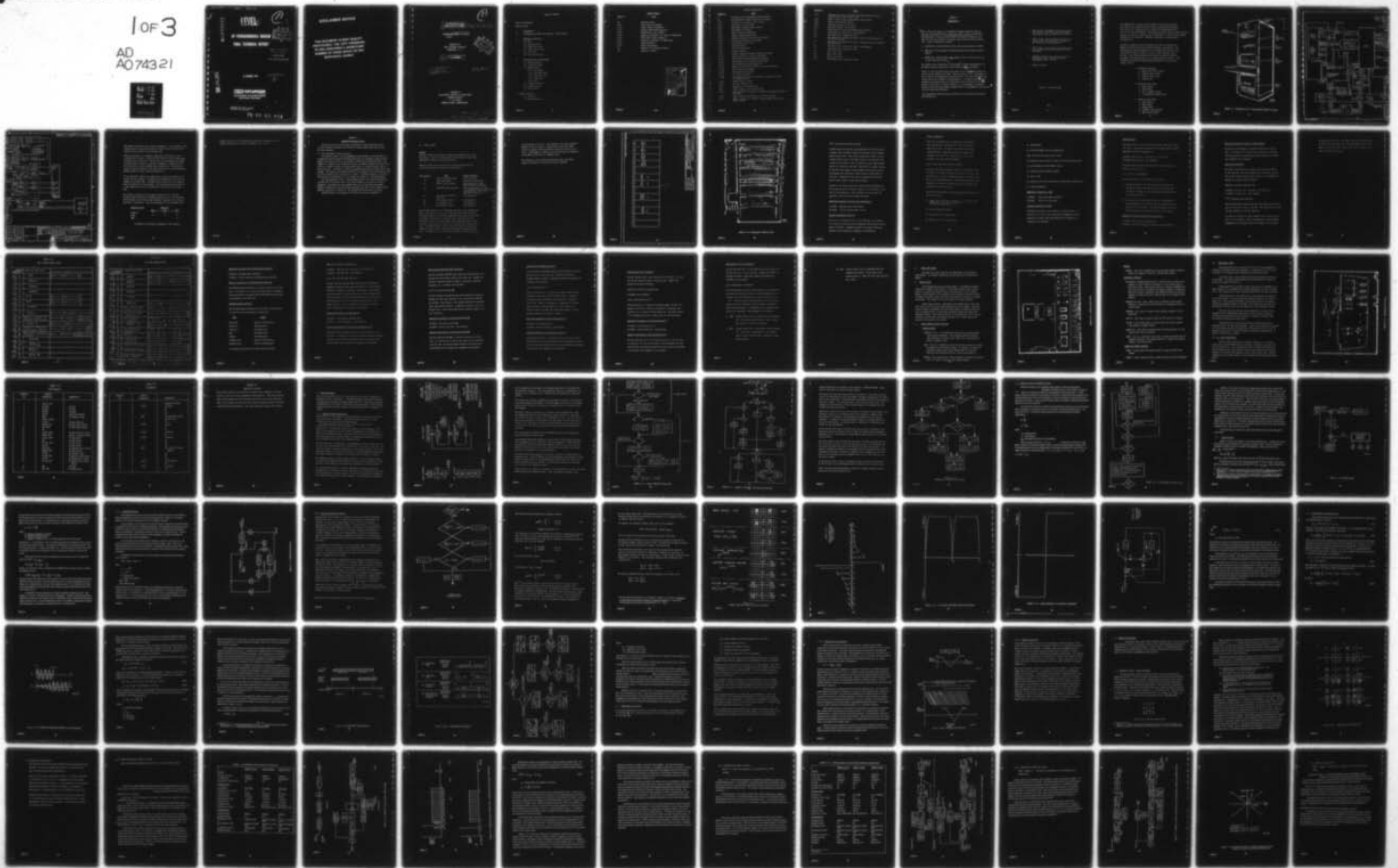
GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 9/2  
HF PROGRAMMABLE MODEM.(U)  
OCT 74

N00014-74-C-0299  
NL

UNCLASSIFIED

1 OF 3

AD  
A074321





011256

7 Copies 4/1/76

AD A 074321

**LEVEL**

C

**HF PROGRAMMABLE MODEM  
FINAL TECHNICAL REPORT**

DDC  
RECEIVED  
SEP 26 1979  
E

THE RUTH H. HOOKER  
TECHNICAL LIBRARY

JUN 28 1978

NAVAL RESEARCH LABORATORY

15 OCTOBER 1974

519,885  
(2)

DDC FILE COPY

**GTE SYLVANIA**  
INCORPORATED  
**ELECTRONIC SYSTEMS GROUP**  
**EASTERN DIVISION**

APPROVED FOR PUBLIC RELEASE  
DISTRIBUTION UNLIMITED

79 09 25 051

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DDC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**

1

6 HF PROGRAMMABLE MODEM .

9 FINAL TECHNICAL REPORT. 15 Apr-15 Oct 74.

CONTRACT NO. N00014-74-C-0299  
15

PREPARED FOR  
NAVAL RESEARCH LABORATORY  
WASHINGTON, D. C.

DDC  
RECEIVED  
SEP 26 1979  
E

11 15 OCTOBER 1974

12 196p.

519,885

PREPARED BY  
GTE SYLVANIA ELECTRONIC SYSTEMS GROUP  
EASTERN DIVISION  
77 "A" STREET  
NEEDHAM HEIGHTS, MASSACHUSETTS

406 451 LAR  
79 09 25 051

## TABLE OF CONTENTS

### LIST OF ILLUSTRATION

### LIST OF TABLES

1. INTRODUCTION
  - 1.1 DELIVERABLE EQUIPMENT AND SOFTWARE - BLOCK DIAGRAM
2. SUBSYSTEM DESCRIPTION
  - 2.1 PSP
  - 2.2 Peripheral Nest
  - 2.3 Paper Tape Reader
  - 2.4 Modem Control Panel
  - 2.5 I/O Panel
  - 2.6 Power Distribution
3. HF MODEM SOFTWARE DEVELOPMENT
  - 3.1 Modem Techniques
  - 3.2 Coding Techniques
  - 3.3 Modem Programs
    - 3.3.1 2400 TDPSK
    - 3.3.2 3600/4800 TDPSK
    - 3.3.3 2400/3600 Codem
    - 3.3.4 4800 Codem
    - 3.3.5 DEFT
  - 3.4 Software Operation
  - 3.5 Program Timing
  - 3.6 Test Results
4. Support Programs
  - 4.1 PSD-PDP-11
  - 4.2 Test Diagnostics

# LIST OF TABLES

TABLE NO.

TITLE

1-1	Delivered Items
2.2-1	Input Channel Assignment
2.2-2	Output Channel Assignments
3.3-1	Uncoded TDPSK Modem Parameters
3.3-2	TDPSK/Product Code (CODEM)
3.3-3	Frequency Assignments for 4800 b/s CODEM Modem
3.3-4	FDPSK (DEFT) Modem Parameters
3.3-5	Frequency Assignment for FDPSK Modem
3.5-1	Timing Estimates
3.5-2	Memory Requirements
4-1	PSP-PDP 11/45 Interface Software
4-2	Addresses of NOP's

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Availand/or special
A-23	4.

## LIST OF ILLUSTRATIONS

<u>FIGURE NO.</u>	<u>TITLE</u>
1-1	Photograph of HF Programmable Modem Processor
1-2	HF Programmable Modem Interface Block Diagram
2.2-1	Location of I/O Boards in the I/O Nest
2.2-2	Photograph of HFM I/O Nest
2.4-1	Photograph of Modem Control Console
2.5-1	Photograph of I/O Panel
3.1-1	Typical Program Organization
3.1-2	Sample Interrupt Signal Flow
3.1-3	Detailed Interrupt Flow Chart for 2400 bps
3.1-4	Output Data Interrupt Flow Diagram
3.1-5	Flow Diagram Radix 4 FFT
3.1-6	4 $\phi$ DPSK Decoder
3.1-7	Amplitude Demodulator
3.1-8	Doppler Tracking
3.1-9	Hilbert Transform Frequency Shifting Procedure
3.1-10	Impulse Response of Hilbert Transformer
3.1-11	FIR Hilbert Transformer Magnitude Response
3.1-12	Phase Response of FIR Hilbert Transformer
3.1-13	Block Diagram Sin/Cosine Generator
3.1-14	Response of Difference Equation to an Input Sinusoid
3.1-15	Data Input Timing Diagram
3.1-16	Synchronization Alignment
3.1-17	Flow Diagram Preamble
3.1-18	Dot Product Amplitude as a Function of Location of Data in Integration Pairs
3.1-19	Preamble Sync Detection
3.2-1	The (9,4) Product Code
3.2-2	Decoding a (9,4) Product Code
3.3-1	TDPSK $\frac{1}{2}$ Duplex Modem, 2400 b/s Mode
3.3-2	Amplitude Versus Frequency Plots-Output Waveforms and Input at FFT Output
3.3-3	TDPSK $\frac{1}{2}$ Duplex Modem, 3600 b/s and 4800 b/s Modes
3.3-4	TDPSK - Product Code (CODEM), $\frac{1}{2}$ Duplex Modem, 2400 b/s and 3600 b/s Modes

FIGURE NO.TITLE

3.3-5	Confidence Levels for Arbitrary Differential Phase Vector in 2-Bit Per Tone (2400 b/s Mode) Case
3.3-6	TDPSK - Product Code $\frac{1}{2}$ Duplex Modem, 4800 b/s Mode
3.3-7	FDPSK $\frac{1}{2}$ Duplex Modem, 4800 b/s
3.6-1	Frequency Response I/O Filter #1
3.6-2	Frequency Response I/O Filter #2
4-1	Block Diagram of Data Paths Between PSP and PDP-11
4-2	Block Diagram of PSP-to-PDP 11/45 Interface
4-3	Data Memory Test Program Flow Chart
4-4	Data Memory Test Program Flow Chart - Test of All DM Locations for Correct Patterns
4-5	Data Memory Test Program Flow Chart - Generation of New Background and Test Patterns
4-6	Data Memory Test Program Flow Chart - Halt Sequence
4-7	Instruction Test Program Flow Chart
4-8	Pass 1 Flow Chart
4-9	Pass 2 Flow Chart
4-10	Pass 3 Flow Chart
4-11	PSP Multiply Test Program Flow Chart

## SECTION 1

### INTRODUCTION

↓ This is the final report of a seven-month program beginning April 15, 1974 and ending approximately October 15, 1974 to design and fabricate two programmable signal processors with the necessary software to provide for real time operation of specific High Frequency Modem configuration. These configurations include:

- a. Conventional time-differential phase shift keyed modulation (TDPSK).
- b. DEFT mode, using frequency differential phase shift keyed modulation (FDPSK).
- c. CODEM mode, combined TDPSK with coding for error control and the use of soft decision decoding.

↖ The contract was accomplished by GTE Sylvania, Needham, Massachusetts for the Naval Research Laboratories under Contract #N00014-74-C-0299.

Table 1-1 lists the deliverable items under the HF Programmable Modem Processor Contract #N00014-74-C-0299. Shown in Figure 1-1 is a photograph of one of the two HF Programmable Modem Processors. The basic subsystems and control panels are labeled on the photograph. An Interface Block Diagram breaking the hardware down into sub-sections and describing the interface between sub-sections is shown in Figure 1-2. The equipment is designed around a basic high speed signal processor which is programmed to provide the real time processing required.

The programmable signal processor delivered was the standard rack mounted GTE Sylvania PSP.

1. Two (2) Each - Programmable Signal Processors, each with 2K x 2K of data and program memory with plug-in expansion capability to 4K x 4K.
2. Two (2) Each - Special purpose circuitry, not a part of the main processor, including two A/D converters for each processor.
3. Two (2) Each - Input devices for reading assembled programs into each Programmable Signal Processor.
4. Interface hardware and software between the Programmable Signal Processors and the Government's PDP 11 DR 11C.
5. Reports and Data -

TABLE 1-1. DELIVERED ITEMS

The standard PSP is located in the middle of the rack and consists of a total of two nests of cards, one nest containing the Arithmetic Unit and one nest containing 2K of 32-Bit Word Program Memory and 2K of 16-Bit Word Data Memory. This memory has been expanded by NRL GFE memory elements to 3K x 3K. Directly below the PSK memory is the Standard Program Console and Writing Shelf. The PSP is described in further detail in Section 2.1.

The PSP is a general purpose processor which must be interfaced to specific controls and devices to meet particular requirements. The requirement in this procurement was for a modem which would be used to perform HF channel testing of a set of voice processors over a HF link. The interface between the PSP and the external device is provided by the Input/Output nest of special purpose circuitry. This circuitry provides the standard modem interfaces to the voice processor, such as

- a. Transmit modem input
  - 1. push-to-talk (PTT) command
  - 2. external frame timing
  - 3. external data timing
  - 4. digital data
- b. Transmit modem output
  - 1. frame timing
  - 2. data timing
  - 3. end of preamble signal
  - 4. rf transmitter keying signal
- c. Receive modem output
  - 1. frame timing
  - 2. data timing
  - 3. digital data
  - 4. preamble tones detected
  - 5. data tones detected

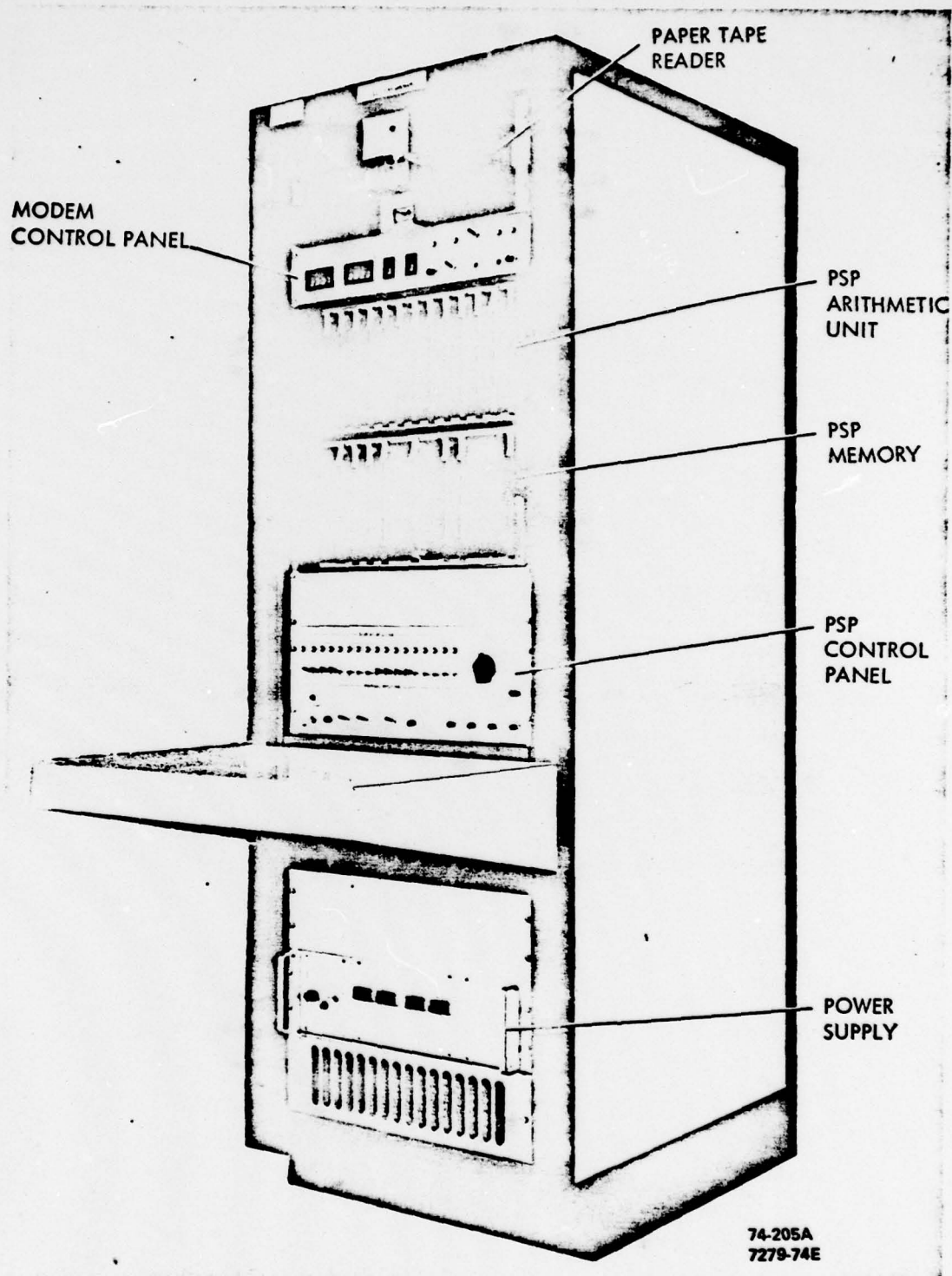
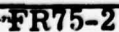
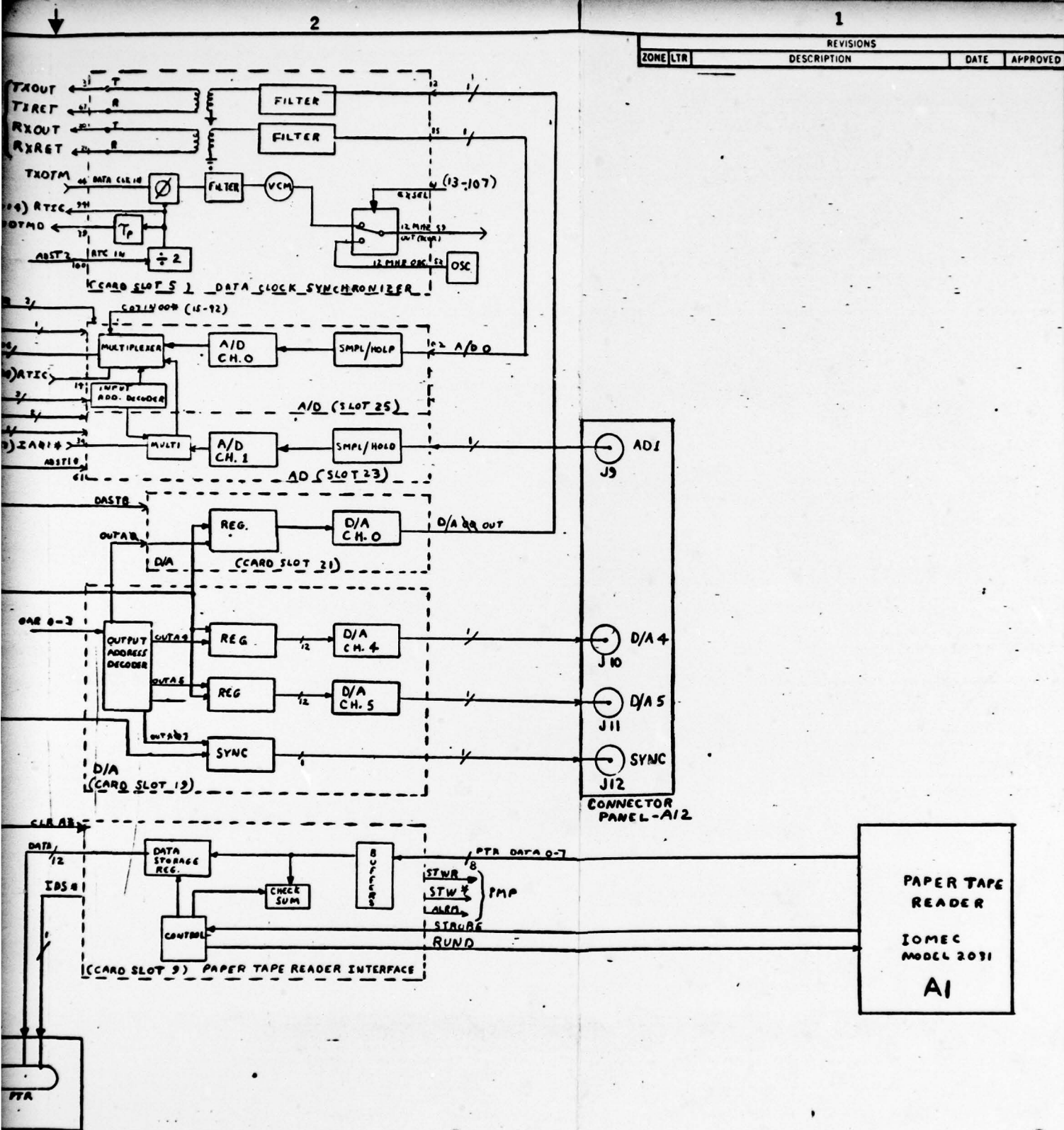


Figure 1-1. Photograph of H. F. Programmable Modem Processor





APPLICABLE DOCUMENTS		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		CONTRACT NO.		<b>GE SYLVANIA INC</b> ELECTRONIC SYSTEMS GROUP - EASTERN DIV NEEDHAM HEIGHTS, MASSACHUSETTS	
TOLERANCES ANGLES ± DECIMALS 2 PLACE ± 3 PLACE ± MATERIAL		SURFACES ✓ MICRO-INCHES		DWN T Hrsat 12 Sept 74 CHK T Hrsat 12 Sept 74 APVD A Rane 9/19/74 APVD J H 20 SEPT RELEASE APVD Hrsat 20 SEPT 74	DATE	BLOCK DIAGRAM H.F.M. I/O NEST	
		FINISH		SIZE	CODE IDENT NO.	DRAWING NO.	
NEXT ASSY	NRI	USED ON	APPLICATION	D	04655	00-593674	SCALE SHEET 1 OF 1

2

A

2

The signals are derived from routines programmed in the Processor in the transmit mode or input data process such that it can be easily interfaced with the Processor in the receive mode.

Since the processor is a computer, means must be provided to input a program when required. Provisions have been provided for three means of input access to each PSP. For the unit in a laboratory environment the hardware and software is provided to assemble a program on a PDP 11/45 and directly read a program into the processor. The ability to directly read data between the processor and the 11/45 is also provided. Further details on this is given in Section 4.0.

For the processor located at a remote site, a paper tape reader is included with the equipment. Programs can be assembled on the 11/45 and punched out on paper tape. The paper tape is then used as an input to read the program into the processor. Also provisions were provided for a card reader which could be interfaced to the processor at a later date.

The software developed for the program included many of the most accepted modem techniques and designs presently in use. Four programs were developed which totaled seven different configurations of data rate and types of modulation. These are summarized in the following chart:

<u>Modulation Type</u>	<u>DATA RATE</u>		
	<u>2400</u>	<u>3600</u>	<u>4800</u>
Kineplex	X	X	X
Codem	X	X	X
DEFT			X

X indicates configuration programmed on this contract

Further details on the Modem Techniques used and a description of the programs developed can be found in Sections 3.2, 3.3, 3.4.

## SECTION II

### SUBSYSTEM EQUIPMENT DESIGN

The following section describes the operation of each functional block in the deliverable equipment and identifies drawing numbers and wirerun lists for each subsystem.

#### 2.1 The Programmable Signal Processor

The central element in the programmable modem hardware is the PSP which is a high-speed digital computer designed specifically to implement complex digital signal processing systems such as modems, vocoders and spectrum analyzers. The computer is composed of an arithmetic section, an address storage and modification section, a control section, an input/output section, a control console, and two memory sections. One memory is used for program storage and the other one for data storage. The memories are designed with TTL, LSI elements and the instruction rate for direct memory access instructions is 4 MHz. A hardware multiply is provided which performs multiplication of two sixteen bit numbers in .750 microseconds.

For this application 2K of 32 bit program memory, 2K of 16 bit data memory, a twelve bit A/D converter and a 12 bit D/A converter is provided with each equipment. The memory has been expanded GFE to 3K of both data and program. Digital interfaces for voice processing equipment and inputs from a modem control operators console are connected to the standard PSP input-output structure.

Further details on the design of the PSP can be found in Appendix C.

## 2.2 HFM I/O NEST

### General

The HFM I/O Nest is physically located in the rear section of the HFM cabinet. The nest contains digital and analogical logic which interfaces the PSP with Peripheral devices.

There are eight logic boards in the I/O nest which perform the following functions as indicated in Figure 2.2-1.

<u>Slot Location</u>	<u>Name</u>	<u>Primary Fuction</u>
5	Data Clock Synchronizer	Phase lock loop
9	Paper Tape Reader	Load PSP Program Memory
13	FDP-11 & 188 Interface	Interfacing PDP-11 with PSP & 188 data transmission
15	PSP Modem Panel Interface	Used by programmer as a tool (1) generating Parameter for real time operations, (2) trouble shotting malfunctions
19	D/A Converter OUTA Ø 4A and Ø 5A	D/A Conversions
21	D/A Converter OUTA Ø	D/A Conversion
23	A/D Converter A/D1	A/D Conversion
24	A/D Converter A/D Ø	A/D Conversion

The I/O Nest consists of an Augat panel rack and a wire wrapped back panel. This chassis contains the connectors that the nest to the peripheral devices, MIL-STD-188 transmission data lines, and the PSP. The chassis is hardwired to the logic nest. The wire run list, #19-593671 defines the signal names on each connector shown in the interface block diagram in Figure 2-2 and defines the wiring to the nest. The signal grounds from the connectors are tied to the nest ground plane by lugs and a ground strap running the length of the nest. A terminal block on the chassis connects +5 volts,  $\pm$  15 volts

and the ground to the nest. Also mounted on the upper connector bracket of the I/O nest is a fixture that houses the 12 MHz oscillator module which provides the timing for the real time clock. A photograph in Figure 2.2-2 shows the physical appearance of the HFM I/O nest and its component parts.

The remainder of this subsection describes each logic board. Applicable documents for the HFM I/O nest: 00-593674.

[illegible]

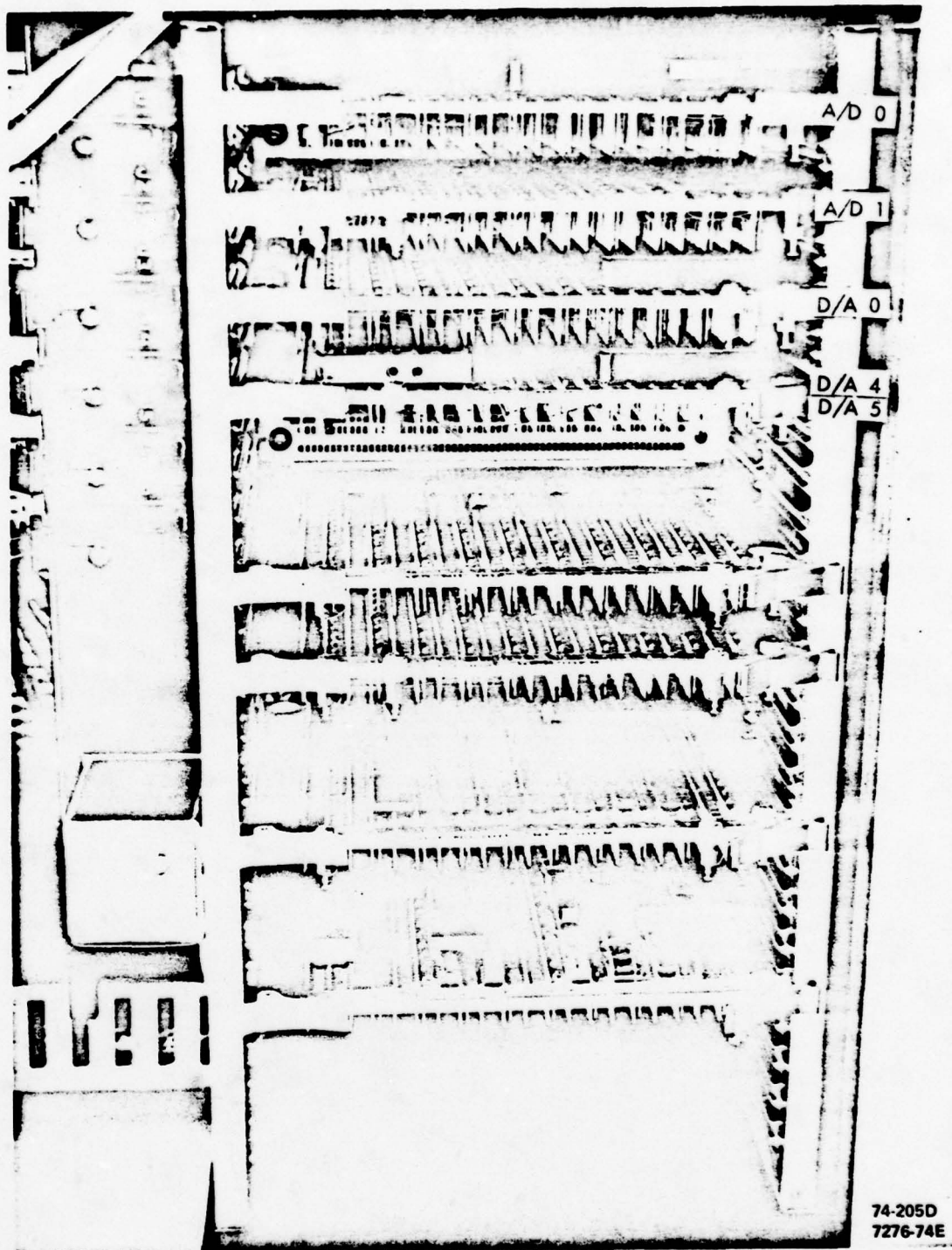


Figure 2.2-2. Photograph of HFM I/O Nest

### 2.2.1 Data Clock Synchronizer (DCS)

The DCS inputs the external clock generated by the data source to phase locks a voltage controlled oscillator using a programmable divider chain. This clock is then used in place of the 12 MHz internal clock as the basic timing to all the real time counters in the Modem. Logic to select the clock under program control is located on this card. The effect is a phase locking of the real time samples to the external clock and also allowing sampling and outputting of data with the correct phase of the clock. Operation in this mode is required when the data source cannot accept a clock from the Modem.

Located on the card is also the circuitry which interfaces the A/D converter and D/A converter channel 0, with the audio interface of the transceiver. This includes an active filter and amplifier chain for both the input and output.

#### Applicable Documents for Data Clock Synchronizer

00-592169 HFM Data Clock Synchronizer

06-592168 Circuit Card Assy HFM - D.C.S.

#### Physical Description of D.C.S.

The D.C.S. is located in slot 5 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits plus several types of discrete components mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the Backside.

### D.C.S. Interfaces

The D.C.S. interfaces with a 12 MHz Osc., D/A converter D/A 0 UTA ~~00~~, A/D Converter A/D "0", the I/O Connector Panel and the PSP. The signals are defined on document number 06-592150. Applicable interface documents are: 19-593671, wire run list - I/O Nest to "J" conn. 1-17 19-593672, wire run list I/O internerst.

#### 2.2.2 Paper Tape Reader Interface (PTRI)

The Paper Tape Reader Interface logic card temporarily stores and buffers tape data, performs a check sum of the data, and provides all the control functions necessary to transfer tape data to the PSP program memory. The paper tape reader is not connected as an input device and can only load memory when the PSP is in the non-run mode.

The Operational Procedure for acquisitioning tape data by the PSP is as follows.

- a) Jumper cable "W18" must be attached to J5 and J6 on the photograph of I/O panel, Figure 3.5-1. I/O Connector Panel (see photo).
- b) Turn on Paper Tape Reader.
- c) Load tape into the tape reader.
- d) PSP must be in the non-run mode.

e) Program CLRA.

f) Push PTR "START" button on Modem Panel

Tape should run and stop at end of tape.

The conditions under which the tape can stop are the following:

a) By depressing the PTR "START" button.

b) Detecting faulty message preamble.

c) End of tape.

d) Incorrect sum check (alarm light on modem panel illuminates).

e) CLRA (programmed).

#### Applicable Documents for PTRI

00-583913 Paper Tape Reader Interface

06-583914 PTRI Circuit Card Assy

#### Physical Description of PTRI

The PTRI is located in slot location 5 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the backside.

### PTPR Interfaces

The PTRI interfaces with the PSP and the Iomec tape reader interface logic. The signals are defined on document number 06-583914.

Applicable interface documents are:

19-593671; Wire run list - I/O nest to "J" conn. 1-17

19-593672; Wire run list - I/O internest

The Paper Tape Data flow from the PTR to the PSP is shown as a dotted line on figure 2-1.

### 3.2.3 PDP-11 & 188 INTERFACE

The PDP-11 Interface card performs two functions.

- 1) It provides drivers for data to and from the PDP-11 and PSP polling address, and performs control functions with the PDP-11. Channel 11 has been assigned for polling data to and from the PDP-11.
- 2) It provides level converters buffers for the MIL-STD-188 data transmission lines, decodes the PSP output address and buffers the output strobe. The data transmission lines are further defined in the I/O panel section of this report.

### Applicable Documents for PDP-11 & 188 Interface

00-592151 PDP Interface & 188 Interface

06-592150 Circuit Card Assy PDP Interface and 188 Interface

### Physical Description of PDP-11 & 188 Interface

The PDD-11 & 188 Interface is located in slot 13 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the backside.

### PDP-11 & 188 Interface

The PDD-11 interfaces with the PSP on one end and the PDP-11 on the other end. The 188 interfaces with the PSP on one end, and the I/O connector panel on the other end. The signals are defined on document number 06-592150.

Applicable interface documents are:

- 19-593671 Wire Run List - I/O nest to "J" Conn. 1-17
- 19-593672 Wire Run List - I/O interconnect

### **2.2.4 PSP Modem Panel Interface**

The PSP Modem Panel Interface provides data selector/multiplexer logic for the flow of data to the PSP. The following chart lists the data channeled to the PSP input bus.

The PSP has 16 input and output channels used to get data into and out of the machine. Data is multiplexed by having channel assignments and using particular bit locations in the channels

for particular data bits. The input output nest then processes the data. The particular input channel assignment used in the HFTM is shown in the Table 2.2-1 and the output channel assignment is shown in Table 2.2-2.

Table 2.2-2

## NRL - OUTPUT CHANNEL FORMAT

CHANNEL		DESCRIPTION	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CHANNEL	HEX		
0	0	D/A	MSB ← DATA → LSB
1	1	SPARE	
2	2	SPARE	
3	3	SPARE	
4	4	D/A	MSB ← DATA → LSB
5	5	D/A	MSB ← DATA → LSB
6	6	SPARE	
7	7	SYNC	← NOT USED →
8	8	RCV DATA, DATA-TIMING FRAME-TIMING XH DATA, XM FRAME	← NOT USED → XMF XMD XDF XDM XLF XLD
9	9	PDP-11	MSB ← DATA → LSB
10	A	SELECT CONTROL EXT. DATA, INTER.	← NOT USED → SSEL SLE
11	B	RF KEY, EN. PREAMBLE KG ENAB., DATA PRESENCE & LIGHT	← NOT USED → TXM TXL SZL SZR DDF DDL XDF XDL TXM TXL
12	C	SPARE	
13	D	RTC 2	
14	E	RTC 1	
15	F	RTC 0	

Table 2.2-1

## NRL INPUT CHANNEL FORMAT

	CHANNEL	DESCRIPTION	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
DEC CHNL	HEX																	
0	0	A/D1	M S B ← DATA → L S B															
1	1	A/D2	M S B ← DATA → L S B															
2	2	SPARE																
3	3	SPARE																
4	4	SPARE																
5	5	DOPLER	← NOT USED → D O															
6	6	RTC PROGRAM INTER. RESET	← NOT USED →															
7	7	PUSH TO TALK (SWITCH)	← NOT USED → P T O															
8	8	VARIABLE 1 SWITCH	CHARACTER 4				3				2				1			
9	9	PDP-11	M S B ← DATA → L S B															
10	A	VARIABLE 2 (SWITCH)	CHARACTER 4				CHARACTER 3				CHARACTER 2				CHARACTER 1			
11	B	MODE (SWITCH)	← NOT USED → CHARACTER															
12	C	DISPLAY (SWITCH)	← NOT USED → CHARACTER															
13	D	EXTERNAL DATA	← NOT USED → D O															
14	E	EXTERNAL FRAME	← NOT USED → F O															
15	F	ROM RESET	← NOT USED →															

### Applicable Documents for PSP Modem Panel Interface

00-592156 PSP Modem Panel Interface

06-592155 Circuit Card Assy. PSP Modem Panel Interface.

### Physical Description of PSP Modem Panel Interface

The PSP Modem Panel Interface is located in slot 15 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the back side.

### PSP Modem Panel Interfaces

The PSP Modem Panel interfaces with the PSP on one end and on the other end it receives data as shown below.

<u>TYPE</u>	<u>SOURCE</u>
PDP-11 OUT	PDP-11 & 188 Interface
Variable 1	PSP Modem Panel
Variable 2	PSP Modem Panel
Mode	PSP Modem Panel
Display	PSP Modem Panel
External Data	PDP-11 & 188 Interface
External Frame	PDP-11 & 188 Interface

The signals are defined on document number 06-592155.

Applicable interface documents are:

19-593671 Wire Run list - I/O next to "J" Conn. 1-17

19-593672 Wire Run list - I/O internerst

#### 2.2.5 D/A Converter OUTA $\phi^{4A}$ & $\phi^{5A}$

The D/A Converter OUTA  $\phi^{4A}$  &  $\phi^{5A}$  card provides buffering and temporary storage for PSP output data OBR 4 to 15 when address OUTA  $\phi^{4A}$  or OUTA  $\phi^{5A}$  is selected. The outputs of the storage registers channel the twelve data bits to their respective D/A converter module. D/A  $\phi^4$  OUT and DA  $\phi^5$  OUT are brought out to BINC Connectors J10 & J11 on the I/O Connector Panel. The primary purpose of these converters are to provide displays since no filtering of the signals are provided.

#### Applicable Documents for D/A Converter

00-592159 D/A Converter OUTA  $\phi^{4A}$  &  $\phi^{5A}$

06-592157 Circuit Card Assy. D/A Converter

#### Physical Description of D/A Converter OUTA $\phi^{4A}$ & $\phi^{5A}$

The D/A Converter OUTA  $\phi^{4A}$  &  $\phi^{5A}$  is located in slot 19 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits, and two 32 pin D/A modules mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the backside.

### D/A Converter OUTA $\phi$ 4A & $\phi$ 5A Interfaces

The D/A converter OUTA  $\phi$ 4A &  $\phi$ 5A interfaces with the PSP on one end and the I/O Connector Panel on the other end. Signals are defined on document number 06-592157. Applicable interface documents are: 19-593671 and 19-593672.

#### 2.2.6 D/A Converter OUTA $\phi\phi$ A

The D/A Converter OUTA  $\phi\phi$  card provide buffering and temporary storage for PSP output data OBR 4 to 15 when address OUTA  $\phi\phi$  is selected. The outputs of the storage register are channeled into a D/A Converter module. DA  $\phi\phi$  out is routed to the D.C.S. TX OUT filter. This output provides the baseband signal out to the transceiver.

#### Applicable Documents for D/A Converter OUTA $\phi\phi$

00-592158 D/A Converter OUTA  $\phi\phi$ A

06-592157 Circuit Card Assy - D/A Converter

#### Physical Description of D/A Converter OUTA $\phi\phi$ A

The D/A Converter OUTA  $\phi\phi$ A is located in slot 21 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits, and a 32 pin D/A module mounted on an Augat board type 8136-RG5 with 30 gauge wire wrapping on the backside.

### D/A Converter OUTA $\phi\phi$ A Interfaces

The D/A Converter OUTA  $\phi\phi$ A interfaces with the PSP on one end and the D.C.S. on the other end. Signals are defined on document number 06-592157. Applicable interface documents are: 19-593671, and 19,593672.

### 2.2.7 Converter A/D "1"

The A/D Converter A/D "1" converts the analog signals routed through J9 located on the I/O Connector Panel. Segments of the analog signal are sampled, and converted to a digital 15 bit word. After the conversion is completed, the digital word is routed to the PSP input bus. When channel 1 is polled the data is stored into the PSP data memory. For the present programs, this input is a spare.

### Applicable Documents for A/D Converter A/D "1"

00-592154 A/D Converter A/D 1

00-592152 Circuit Card Assy. A/D Converter

### Physical Description of A/D Converter A/D "1"

The A/D Converter A/D 1 is located in slot 23 of the I/O nest. It consists of 14- and 16- pin dual-in-line integrated circuits and two 32 pin A/D modules mounted on an Augat board type 816-RG5 with 30 gauge wire wrapping on the backside.

### A/D Converter A/D 1 Interfaces

The A/D Converter A/D 1 card interfaces with the PSP on one end and the I/O Connector panel on the other end. Signals are defined on document 06-592152.

Applicable interface documents are:

19-593671, and 19-593672.

### 2.2.8 A/D Converter A/D "0"

The A/D Converter "0" converts the analog signal "RX OUT" into digital 15 bit word. After the conversion is completed, the digital word is routed to the PSP input bus. Each time channel "0" is addressed the data is stored into the PSP data memory.

### Applicable Documents for A/D Converter A/D "0"

00-592153 A/D Converter A/D "0"

00-592152 Circuit Card Assy. A/D Converter.

### Physical Description of A/D Converter A/D "0"

The A/D Converter A/D "0" is located in slot 25 of the I/O nest. It consists of 14- and 16- pin dual in line integrated circuits and two 32 pin A/D modules mounted on an Augat board type 816-RG5 with 30 gauge wire wrapping on the backside.

### A/D Converter A/D "0" Interfaces

The A/D Converter A/D "0" card interfaces with the PSP on one end and the D.C.S. on the other end. Signals are defined on document 06-592152. Applicable interface documents are: 19-593671, and 19-593672.

#### 2.2.9 Assignment of Interrupts

The interrupt and real time counter structure is the heart of any real time processing system. Further explanation of the software use of the interrupt logic can be found in section

3.1 Modem Techniques and Algorithms. The I/O circuitry defines which A/D converter or input channel is assigned to which real time counter. The HFPD has three interrupt logics and two A/D converters. The assignment is as follows:

- 1) RTC0 (Output address "F") is associated with A/D address "0" - it has the highest priority - (PC = 0002).  
Input "0" will reset this interrupt.
- 2) RTC1 (Output address "E") is associated with A/D address "1". It has the lowest priority - (PC = 0000).  
Input "1" will reset the flop. (This A/D is spare at the moment.)

- 3) RTC2 (Output address "D") is associated with the external data timing. It has next to the highest priority. Input "6" will reset the flop (PC = 0001).

### 2.3 PAPER TAPE READER

The paper tape reader used was the IQMEC Model 2031 Perforated Tape Handler. For further information see the Operation and Maintenance Manual.

### 2.4 CONTROL PANEL

The configuration has two control panels - a standard PSP program console and a modem control console. The program console allows the program to be started and stopped; words to be read into or out of program memory, or data memory, or the register field to display the major internal registers and control flops; and allows the operator to single step through the program. The primary use of this console is to debug programs.

Figure 2.4-1 contains a photograph of the modem control panel thru which most of the real-time program modifications will be made. The modem control console is segmented into four major groups--a program control section, a display section, a data/radio interface section and a paper tape interface section. The real-time control of the program and all of the connections to the HF radio, the display devices, and the data sources and sinks can be effected at this panel.

#### 2.4.1 MODEM CONTROL CONSOLE FUNCTIONS

##### PROGRAM CONTROL

- VARIABLE 1 and 2 - Each thumbwheel switch will be used to modify 16-bit program parameter. The parameters modified could be such parameters as sampling rate, frame rate, synchronization speed, or Doppler speed.
- MODE - The 4-bit thumbwheel switch will be used to control the mode of operation of each modem program. For example, six modes of the TDPSK program would be half-duplex 2400 b/s, 3600 b/s, 4800 b/s and full-duplex 2400 b/s, 3600 b/s and 4800 b/s.
- DOPPLER - This switch will be used to disable the Doppler correction.

This is a feature common to many hardware modems.

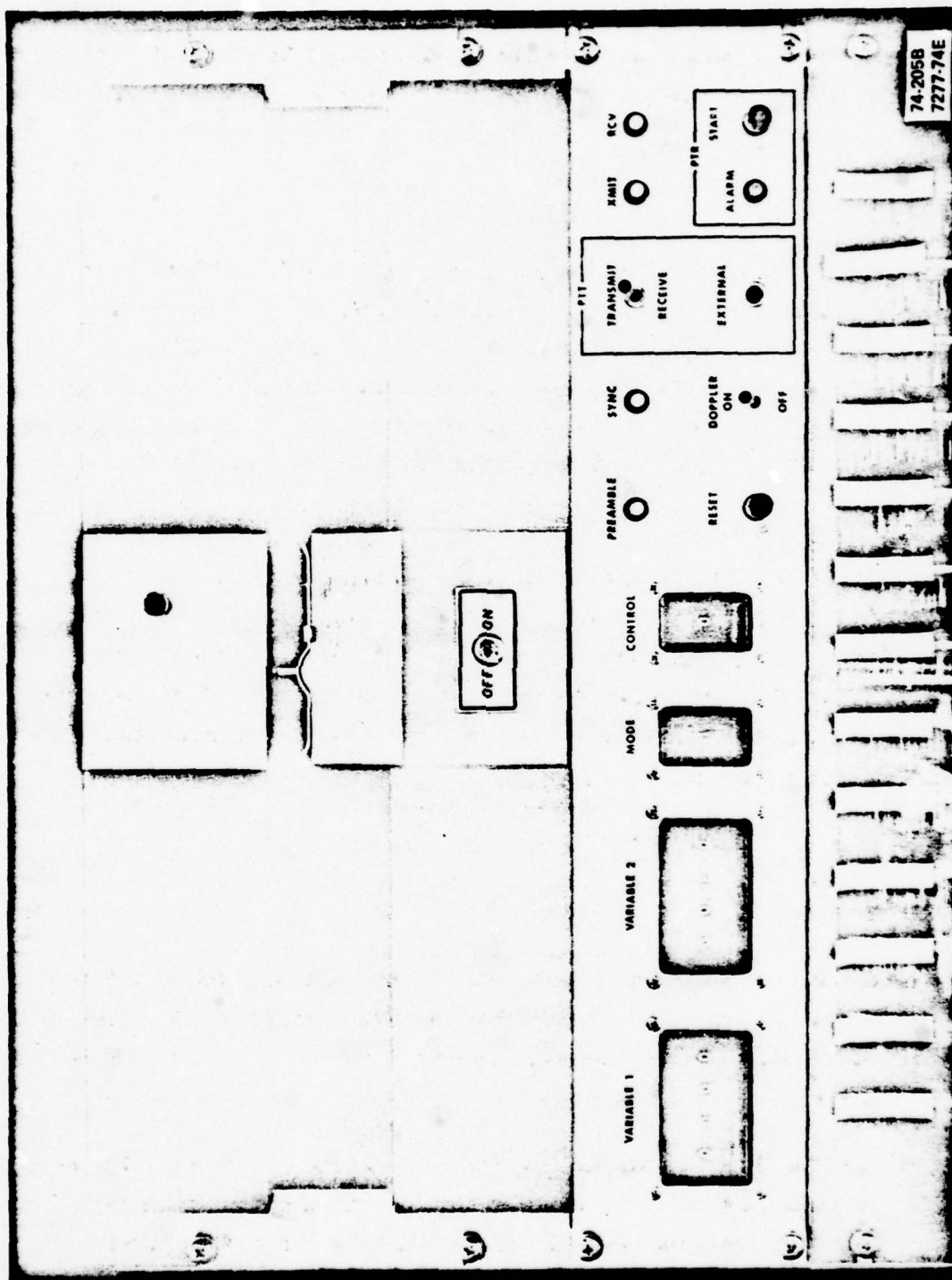


Figure 2.4-1. Modem Control Panel

### DISPLAY

- **CONTROL** - This 4-bit thumbwheel will control which display functions and/or sync appears at the D/A1, D/A2 and SYNC outputs.

### DATA/RADIO INTERFACE

- **PTT CONTROL; TRANSMIT/RECEIVE Switch** - The PTT switch is used by the program to control whether the modem is in the transmit mode or the receive mode. When the PTT switch is turned on (TRANSMIT position) the receiver is disabled and transmit preamble is generated. When this switch is returned to the RECEIVE position the transmit program is turned off and the receiver goes into a preamble search routine.
- **EXTERNAL PTT JACK** - When a phone jack is inserted, the PTT switch function is disabled and the PTT function is accepted from an external source.
- **PREAMBLE** - This light is turned on when transmit preamble is being generated.
- **XMIT ON** - This lamp is turned on when the PTT function is enabled.
- **IN SYNC** - The IN SYNC light is turned on when the receive program has acquired a preamble signal.
- **RECEIVE ON** - This lamp is turned on when the modem program is in the receive mode (PTT not enabled).
- **RESET** - This is a momentary switch used to restart the modem from the preamble sequence in the transmit mode or to restart the preamble search in the receive mode.

### PAPER TAPE READER INTERFACE

- **START** - A push button that either starts or stops the paper tape reader.
- **ALARM** - A light indicating when a check sum error has been recognized.

## 2.5 INPUT/OUTPUT PANEL

The input/output panel located on the rear of the rack provides all electrical interfaces with other equipments. A photograph of the I/O panel in Figure 2.5-1 illustrates the connections and controls available.

A.C. power input, circuit breakers and fuse and located to the left on the panel. A circuit diagram showing the interconnection of these controls can be found in Section 3.6.

Interconnections to the NBT or other data source are provided on the Data connector. Pin assignments and drawing nomenclature are given in Table 2.5-1. All signals are  $\pm 6$  Volts defined by MIL-STD-188 low level interface.

Interconnections to the transceiver and key generator are provided on the RF connector. Pin assignments and drawing nomenclature are given in Table 2.5-6. The transmit and receive audio output tones are transformer coupled and impedance matched to 600 Ohms. The transmitter key signal and key generator control signal are MIL-STD-188 low level compatible signals.

The test signals are brought to BNC connectors for access to test equipment such as oscilloscope and counters. These include the Analogue to Digital Spare Converters, the two Digital to Analogue Converters used for software display and the Sync Pulse.

Interconnections to the PDP-11/45 DR11-C is made thru the three connectors PDP-11 IN, OUT and Card Reader. Three 25 foot cables were provided for this purpose with BERG connectors for interconnections with the DR 11C interface of the PDP-11. Cable information is provided in drawing Nos. 19-592214, 19-59-2215 and 19-592216.

## 2.6 A.C. POWER DISTRIBUTION

The A.C. Power distribution is shown in Figure 2.6-1. The prime power comes to the processor thru the I/O panel located on the back of the rack on Connector J1. It is controlled by the A.C. Power on-off switch on the PSP control panel which switches a relay routing power thru two circuit breakers to the Launch Power supplier and A.C. Strip on the rack. Further information on the Launch Supplier can be obtained from the Launch Operating Manual.

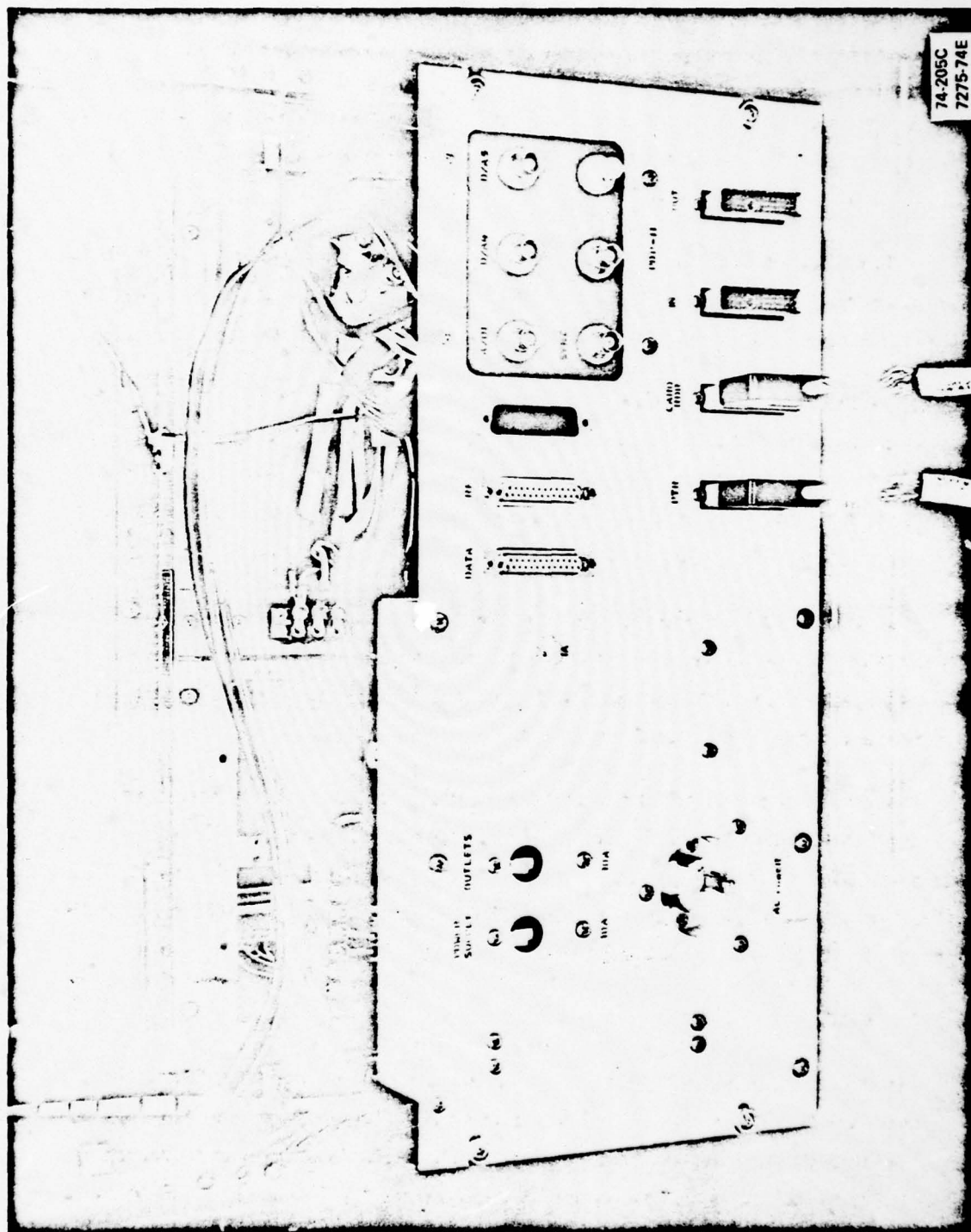


Figure 2.5-1. Photograph of I/O Panel

**TABLE 2.5-1**  
**DATA CONNECTOR**

CONNECTOR PIN	DRAWING DESIGNATION	DESCRIPTION
1	Chassis Gnd	
2	TX Data	TX Data
3	RX Data	RX Data
4	RXD FRM	RX Frame
5	EX PRM	TX End of Preamble
6	EPRM	RX Preamble Detect
7	Signal Ground	
8	D Pres	RX Data Tones Det
9	RXPIM (SEC)	RX Data Timing Return
10	XMB (SEC)	TX Data Timing Return
11	NA	
12	D Pres (SEC)	RX Data Tones Det Return
13	EXPRM (SEC)	TX End of Preamble Return
14	TX Data (SEC)	TX Data Return
15	XMD	TX Data Timing
16	RX Data (SEC)	RX Data Return
17	RX DTM	RX Data Timing
18	EPRM (SEC)	RX Preamble Detect Return
19	RXD FRM (SEC)	RX Frame Return
20	TX FRM	TX External Frame
21	TX FRM (SEC)	TX External Frame Return
22	TX DTM	TX External Data Timing
23	TX DTM (SEC)	TX External Data Timing Return
24	XMF	TX Frame
25	XMF (SEC)	TX Frame Return

TABLE 2.5-2

## RF CONNECTOR

CONNECTOR PIN	DRAWING DESIGNATION	DESCRIPTION
1	-	Protective Ground
2	TX OUT	TX Output
3	-	NA
4	-	NA
5	-	NA
6	KGEN	Key Generator Control
7	-	Signal Ground
8	-	NA
9	TX KEY	TX Key
10	-	NA
11	-	NA
12	RX RET	RX Return
13	-	NA
14	TX RET	TX Return
15	-	NA
16	-	NA
17	-	NA
18	KGEN	Key Generator Control Return
19	-	NA
20	-	NA
21	-	NA
22	TX KEY	TX Key Return
23	-	NA
24	RX OUT	RX Output
25	-	NA

### SECTION III

#### Operational Software

This section describes the operational software which implements the seven modems in real time on the programmable modem processor. The modem technique used in the programs are first discussed followed by the coding techniques. Each of the programs are discussed with pertinent parameters given along with timing and memory requirements. The final subsection contains test results.

### 3.1 Modem Techniques

This section describes the modem algorithms which are used to implement the systems described in section 4.5. The typical program organization utilizes interrupt loops to input and output samples and data. The actual processing accomplished on a baud basis is provided in the main program. The following description provides further detail on the techniques used to implement H.F. modems.

#### 3.1.1 Typical Program Organization

Figure 3.1-1 is a flow chart of a typical program. The routines are broken up into four basic categories: the sample interrupt routine, the data interrupt routine, wait and main loop routines.

The sample interrupt routine, shown in detail in Figure 3.1-2 is entered from the main loop at the system sampling rate. The program counter and contents of the accumulator at the time of the interrupt are saved so that the operation which was being performed in the main loop can be continued at the end of the interrupt routine. The interrupt routine is coded so that the longest path through the routine will be completed before the next interrupt signal.

The interrupt rate is controlled by a programmable real time counter, and can be set to any value from 125 nsec to 512  $\mu$ sec to an accuracy of 125 nsec. The real time counter is usually initialized in the start-up routine. The interrupt pulse strobes the A/D converter which in turn interrupts the program when the input conversion is complete. The output stores a number in a buffer.

The interrupt routine has four primary functions. As the interrupt routine is entered, the program control branches to a line signal calculation routine, where the output sample is calculated and transmitted to the D/A converter.

If a table lookup procedure is used to generate the output tones and the heterodyne tone, registers are used to keep track of the location of each tone in the lookup table. After a sample of the sum of output tones is generated, the output tone register file must be updated. Each register is incremented

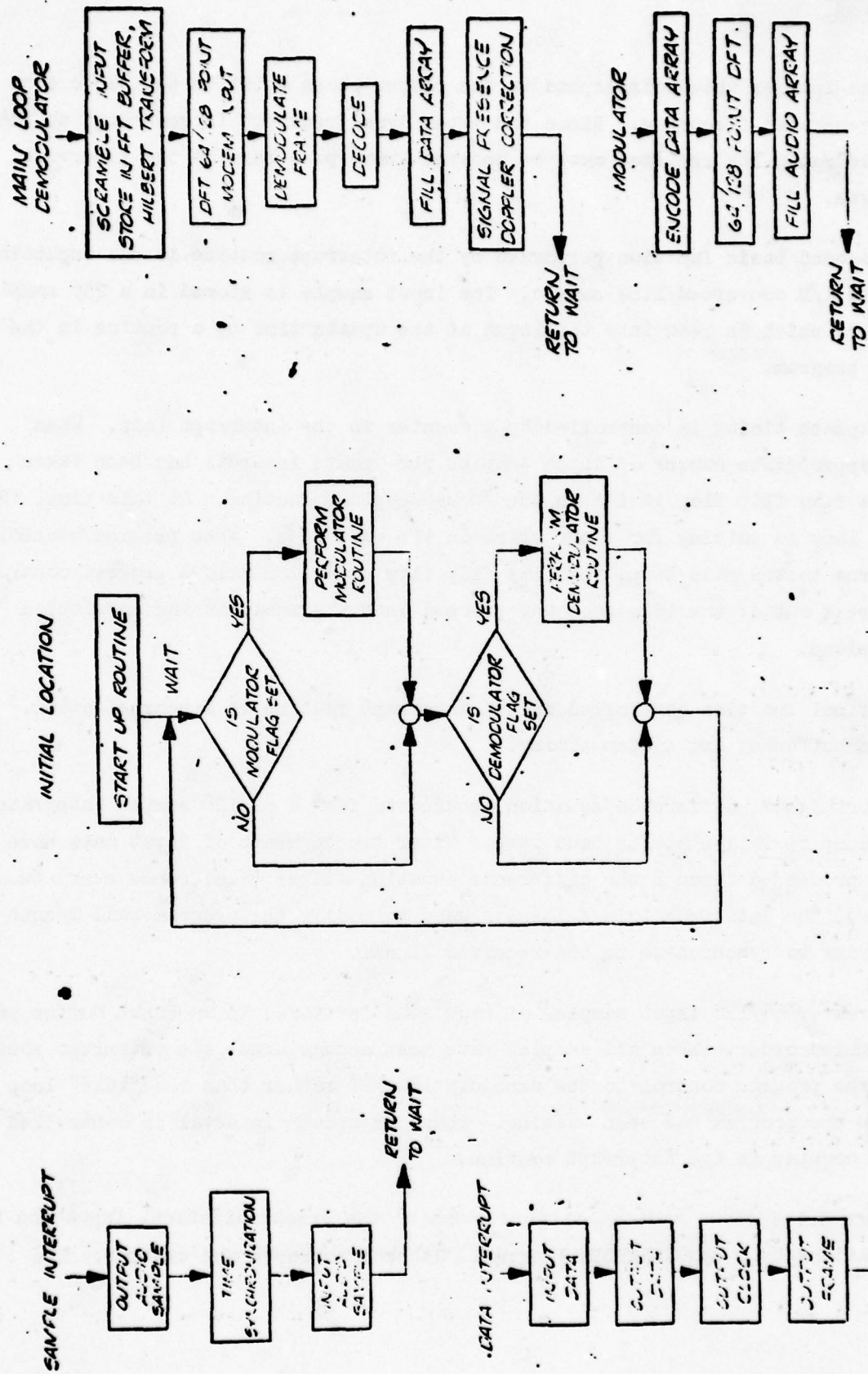


FIGURE 3.1-1 TYPICAL PROGRAM ORGANIZATION.

by the integer which correspond to the proper phase shift in the table for the generated frequency. Since the heterodyne frequency is generated at the sample rate, its register must be updated every pass through the interrupt routine.

The second basis function performed by the interrupt routine is the inputting of the A/D converted line sample. The input sample is stored in a 256 sample buffer, which is read into the input at the update time by a routine in the main program.

The update timing is controlled by a counter in the interrupt loop. When the appropriate number of input samples per update interval has been taken in, a flag flip flop is set in the "housekeeping" routine. At this time, the main loop is waiting for a new start in the main loop. When program control returns to the main loop, the flag flip flop condition allows program control to break out of the idle loop and proceed into the analysis and modulation functions.

The final function controlled by the interrupt routine is synchronization, input buffering and system timing.

One orthogonal difference equation is used to form a  $64/128$  sample integrate and dump operation at the baud rate. After two segments of input data have been processed through the difference equation filter (i.e., once every two bauds), the data from this filter is used to modify the program baud length in order to synchronize to the received signal.

The center  $64/128$  input samples of each baud is stored in an input buffer in scrambled order. When all samples have been accumulated, the interrupt routine returns program control to the demodulation FFT rather than the "idle" loop where the program had been waiting. Thus, an update interval is controlled by a counter in the interrupt routine.

Figure 3.1-3 shows a detailed flow graph of the synchronization, input and timing operations/2400 bps KINEPLEX Program. Other interrupts are similar. The

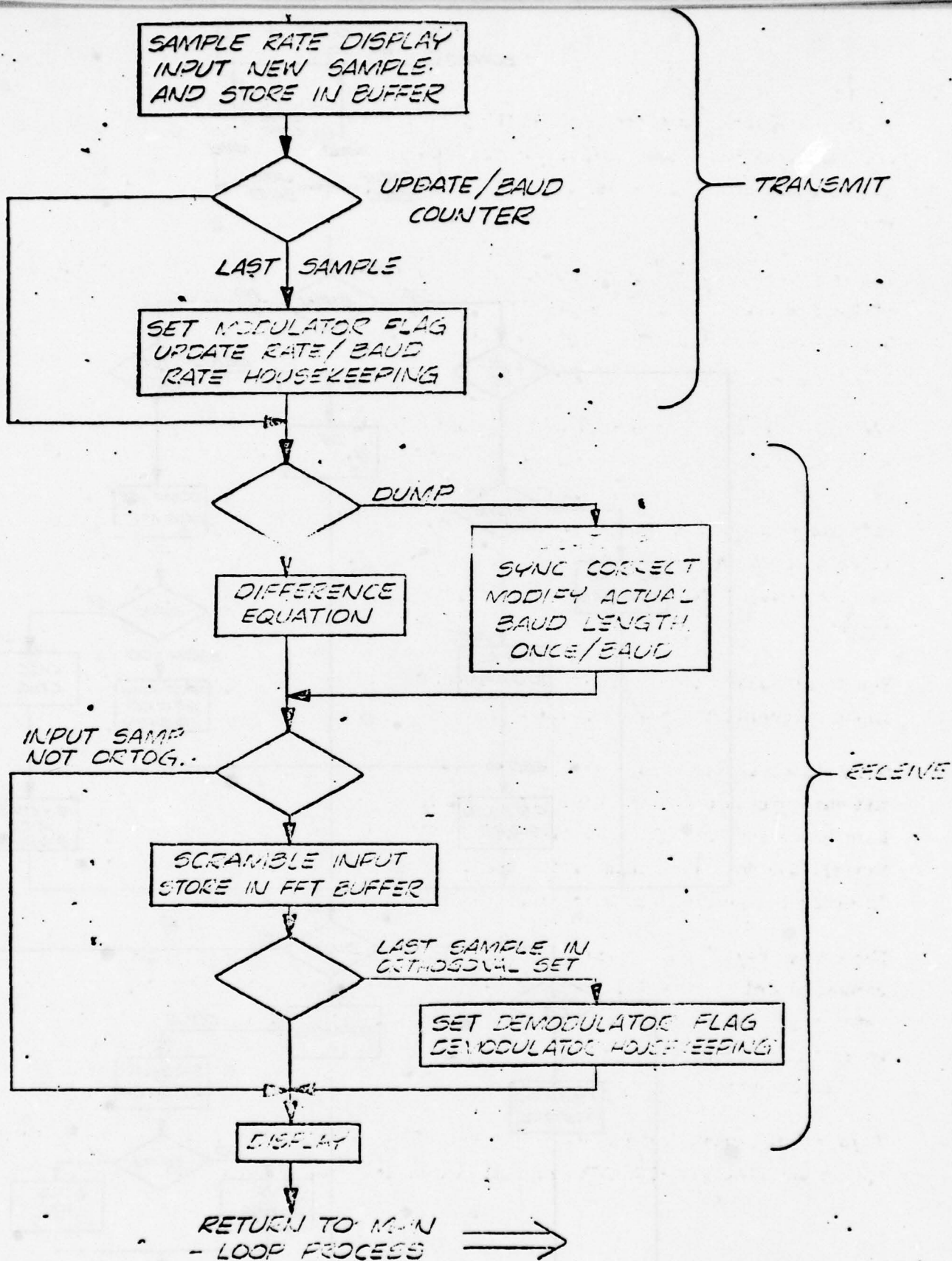
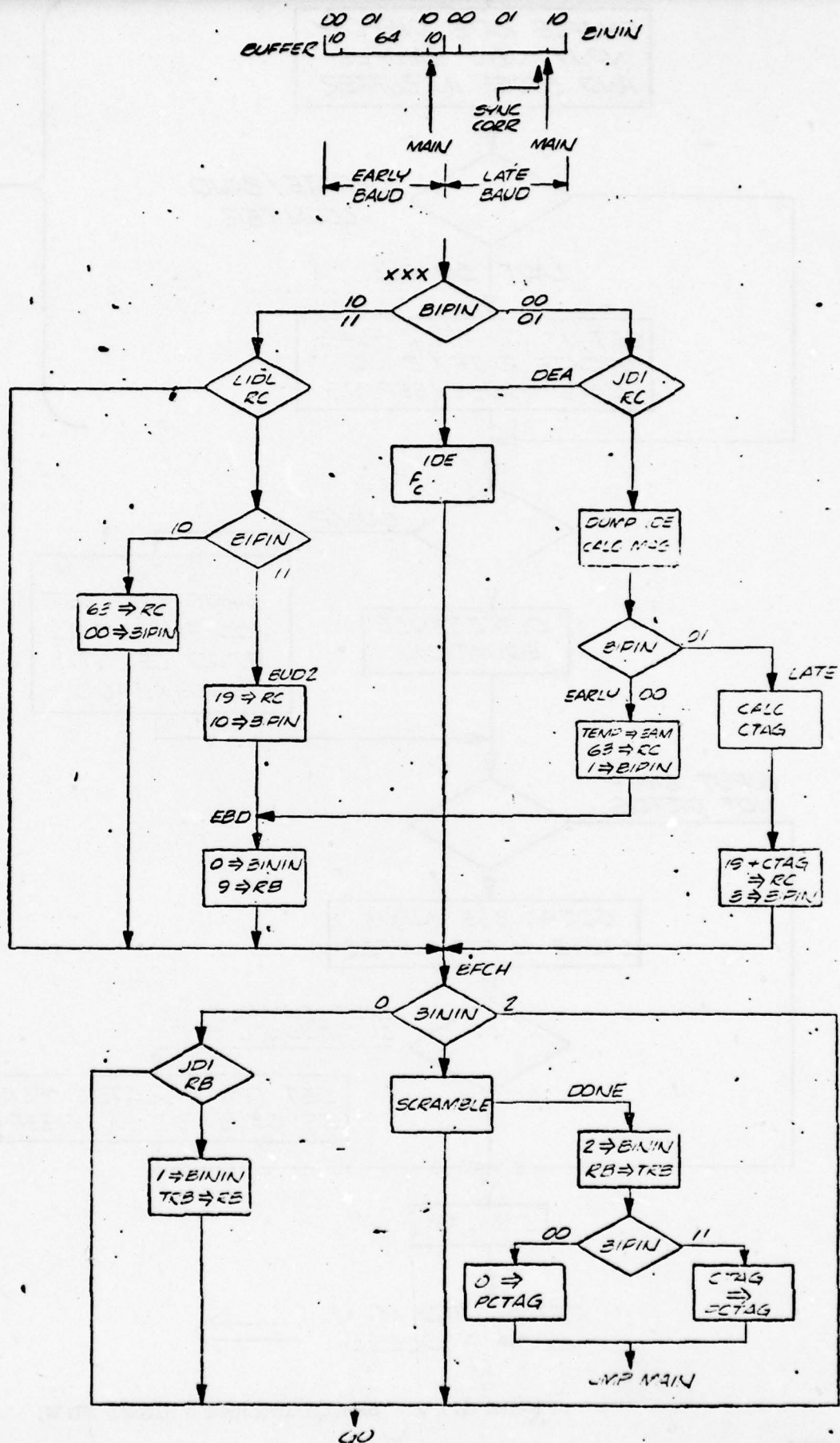


FIGURE 3.1-2 SAMPLE INTERRUPT SIGNAL FLOW.



timing diagram shows the states of two variables - BIPIN and BININ - whose two low order bits are used for timing control.

The output data interrupt loop provides the interface to the data source and sink in the same manner as the audio sample interrupt loop provides the interface to the transceiver. This interrupt loop operates from the second priority real time counter. The interrupt rate is set at twice the data rate. For instance, if the data rate is 4800 bits per second, the interrupt rate will be 9600 interrupts per second.

Referring to figure 3.1-4, upon an interrupt, the program counter jumps to the starting location of the interrupt loop; the A register is saved and the A/D channel 6 is inputted. This enables the interrupt loop for the next interrupt. A decision is then made whether internal or external data is to be used in the program. If internal data is requested, the interrupt loop returns to the main program.

In the external data mode, a decision must then be used to determine if the modem will accept an external clock or output data based upon an internally generated clock. This decision is based upon an input from bit 1 of the mode control switch.

The clock period is then divided into two parts, the period of time when the clock cycle goes from low to high and data is read out of the modem and when the clock cycle goes from high to low and data is read into the modem. In order to determine the phase of the external clock, the recreated clock from the phase locked loop is inputted on channel 2 and is sampled when the software is in the external mode to determine which cycle of the clock is up at the time of sample.

In the internal mode a flag is initialized during the start-up routine and is set and reset in the interrupt routine to keep track of the clock cycle.

Data, clock and framing information is stored in a memory location and outputted once per data interrupt cycle.

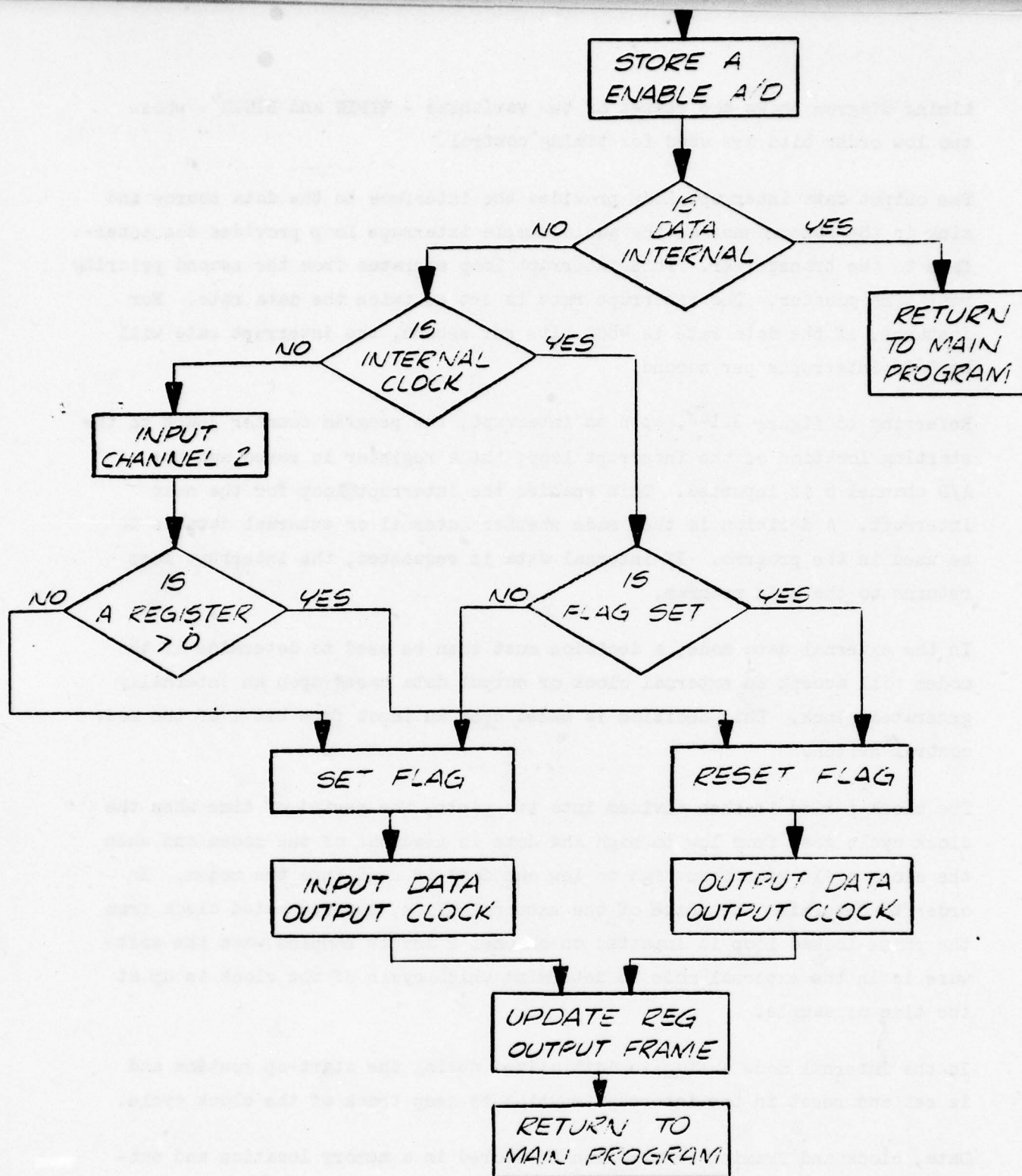


FIGURE 3.1-4  
OUTPUT DATA INTERRUPT FLOW DIAGRAM

### 3.1.2 MODULATION AND DEMODULATION

With the exception of one modulator which utilizes a table lookup algorithm (the 2400 b/s TDPSK mode), the Fast Fourier Transform is used for all modulator synthesis and demodulator analysis operations. A table lookup modulator generates the composite tone set from a cosine table and, although it takes more processing time than an FFT modulator, it has the advantage of generally requiring less program memory. As the number of tones in the set increases a table lookup modulator becomes less and less practical.

For more than ten tones, an FFT is the most efficient method in terms of processing time to synthesize a composite tone set or to analyze a received composite signal. The size of the FFT,  $N$ , must be the lowest power of two which satisfies the following relationships:

$$N = \frac{f_s}{f_t}$$

$$f_s > 2f_{HI}$$

where:

$f_s$  = sampling rate

$f_t$  = tone spacing

$f_{HI}$  = the highest frequency tone transmitted.

For the modem structures developed,  $N$  works out to either 64 or 128. Since a radix-4 FFT requires roughly a factor of two less processing than a radix-2 FFT (1.6 milliseconds versus 3.2 milliseconds for a 64-sample complex FFT on the PSP), the radix-4 algorithm is used exclusively. The flow diagram for a radix 4 FFT is shown in Figure 3.1-5.

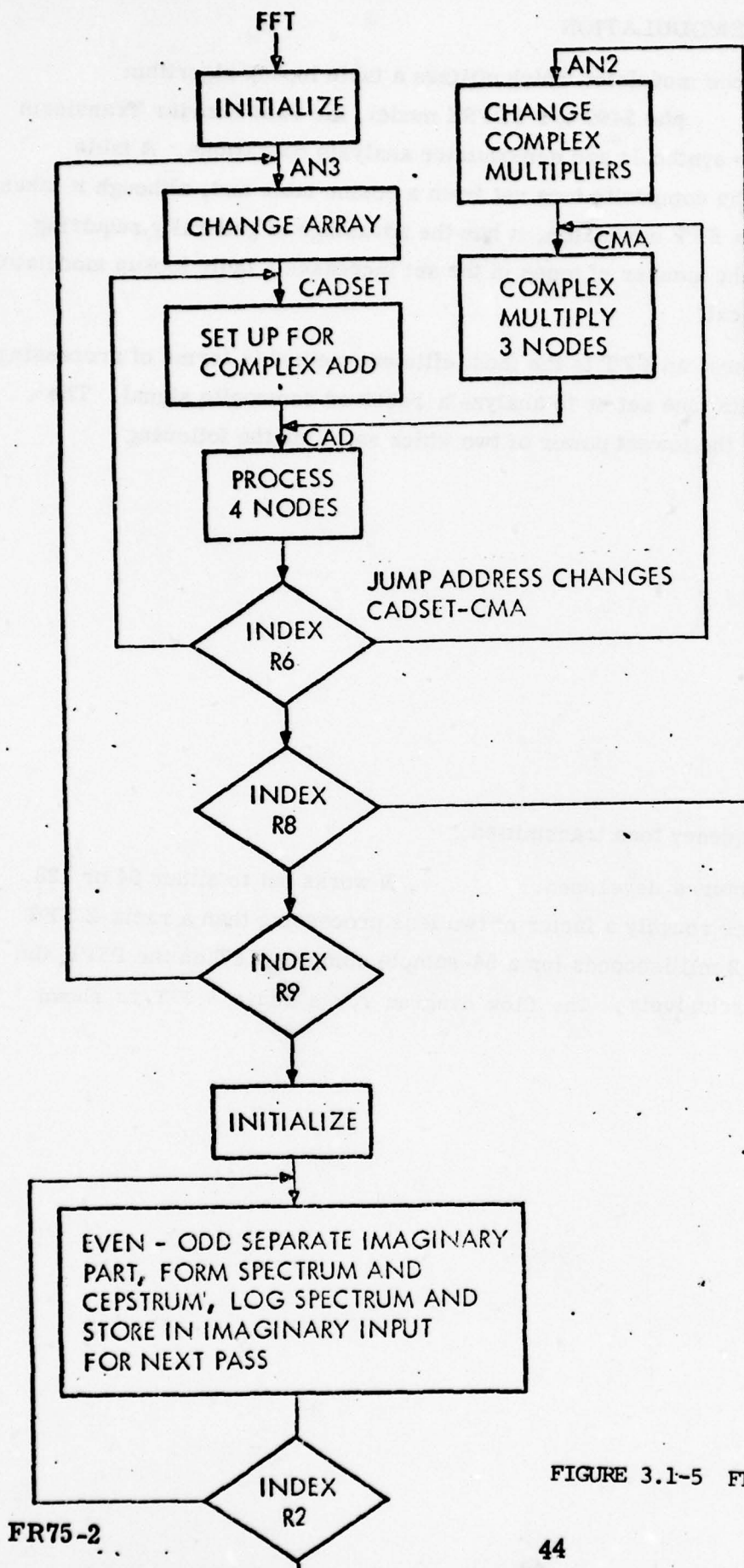


FIGURE 3.1-5 FLOW DIAGRAM OF A RADIX 4 FFT.

Because a 128-sample DFT cannot be implemented directly with a radix-4 FFT (128 is not a power of four), the 128-sample DFT's required will be implemented by a real only transform algorithm which uses a complex 64-sample, radix-4 FFT. The real only transform algorithm loads adjacent samples of the real valued input into the 64 real and 64 imaginary inputs to a 64-sample FFT. After the FFT is performed, the complex output is even-odd separated and weighted by a complex sinusoid to produce the real and imaginary outputs of the 128-sample input sequence.<sup>1</sup> The entire 128-sample real only algorithm can be accomplished in 2.54 milliseconds on the PSP. In addition to the time savings, this algorithm has the advantage of using a common FFT routine in all of the modes.

Although it is not intuitively obvious, the real only algorithm can be used to generate the composite tone set from the complex vectors which define the modulation on each tone. The algorithm<sup>2</sup> uses the symmetric properties of a real transform and is shown in block diagram form in section 2.5. The basis of the algorithm is the fact that a real input must have a spectrum whose real part is even-symmetric about N/2 (64 for this case) and whose imaginary part is odd-symmetric about N/2.

In the following two subsections, phase and amplitude detection are discussed in more detail.

### 3.1.2 . DPSK Detection

Figure 3.1-6 is a block diagram of the 4-phase DPSK decoder. Although the vectors are used as complex quantities in the processor decoder routine, they are expressed in magnitude and phase notation to simplify the explanation. Thus, the input vector,

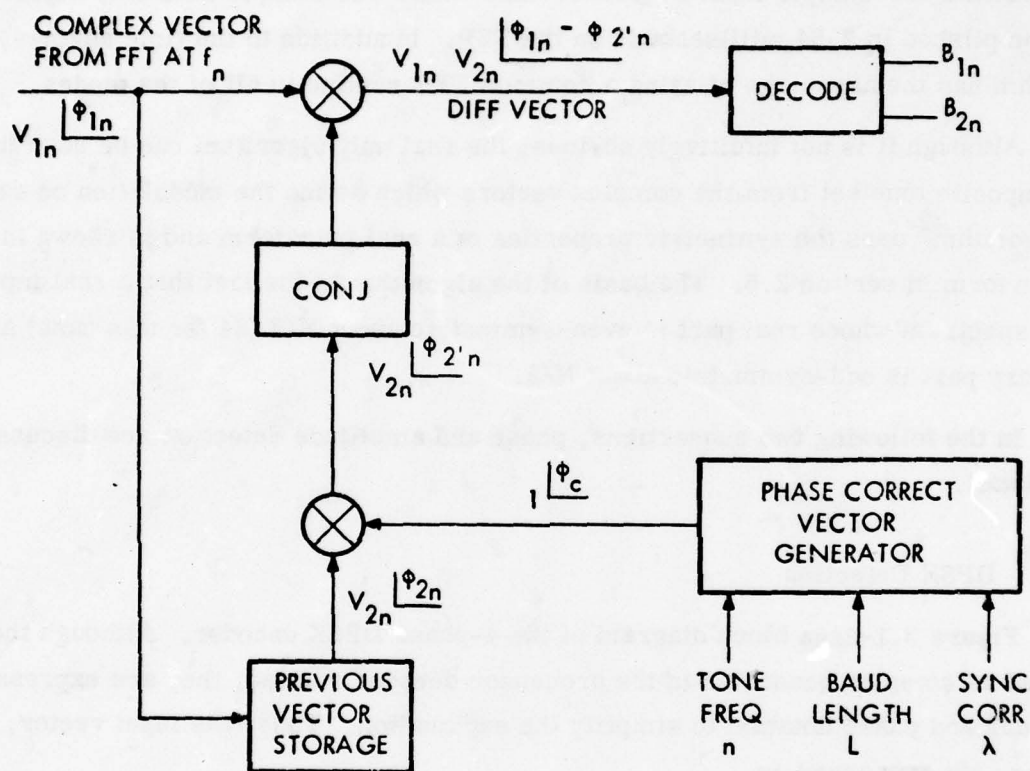
$\mathcal{R}_n - j\mathcal{I}_n$ , is expressed as

$$A_{1n} \angle \phi_{1n} = \mathcal{R}_n - j\mathcal{I}_n$$

where  $A_{1n} \angle \phi_{1n}$  is the present FFT output vector for the  $n^{\text{th}}$  orthogonal modem tone.

To decode the two frame bits associated with the  $n^{\text{th}}$  tone, the phase of this vector must be compared to the phase of the previous  $n^{\text{th}}$  vector,  $A_{2n} \angle \phi_{2n}$ . Since the baud length

1. This operation has been implemented and is fully described in Multi-transform FFT Techniques, J. deLellis, Systems Engineering Technical Memorandum No. 1.
2. This algorithm has been implemented and is described in the Proposal for Goldwine Telephone Subscriber Unit Using Carlos, GTE Sylvania Proposal P71-13, dated 14 May 1971 (Secret-Crypto)



1250-70E

Figure 3.1-6: 4φ DPSK Decoder

is not orthogonal and the Fourier Transform algorithm measures phase only with respect to the first data sample, the phase of the previous vector must be corrected before it is compared with the present vector. A correction vector is generated to properly advance the phase of the previous vector to the same reference sample implied in the present vector:

$$\phi_c = n(L + \lambda) \frac{2\pi}{N},$$

where

$N$  = number of samples in the FFT

$n$  = harmonic number of the tone

$L$  = number of samples/ baud

$\lambda$  = number of samples of sync correction applied since previous vector

In Equation (4-2), the quantity  $2\pi/N$  is the phase angle that the fundamental tone is advance for each single sample advance. This quantity is multiplied by the harmonic number of the tone being processed,  $n$ . Finally, the integer,  $(L + \lambda)$  is the number of samples over which the previous vector must be advanced to have the same phase reference as the present vector.

The previous vector is simply multiplied by the correction vector to compute the corrected vector,  $A_{2n} \mid \phi_{2,n}$ :

$$A_{2n} \mid \phi_{2,n} = A_{2n} \mid \theta_{2n} \times 1 \mid \phi_c$$

Finally, the corrected vector is conjugated and multiplied by the present vector to produce the difference vector

$$A_{1n} A_{2n} \mid \phi_{1n} - \phi_{2,n} = V_{2n} \mid -\phi_{2,n} \times V_{1n} \mid \phi_{1n}$$

The phase angle of the difference vector determines the state of the two data bits associated with the tone being decoded. Since the difference vector is actually represented in the processor as a complex number, the sign of the imaginary part is used to determine the state of the most significant bit and the sign of the real part is used to determine the state of the least significant bit.

In 8 $\phi$  DPSK, the same algorithms are used to generate a difference vector. The change is, of course, that three bits must be demodulated. Detection will be performed by utilizing the initial real and imaginary values of difference vector to determine the quadrant of the received vector, shifting the difference vector by  $\pi/4$  radians, and using the resultant real and imaginary components to specify octant of the received vector.

### 3.1.2 Amplitude Detection

In the high data rate modes of the error control modem programs, two amplitudes will be transmitted on each tone to yield one data bit per tone. Figure 3.1-7 is a block diagram of the technique which will be used for amplitude detection.

The magnitude of the  $n^{\text{th}}$  output slot of the FFT will be compared to a threshold which is initially set to the mean of the expected amplitudes. The demodulation decision for the received bit is based on the sign of the difference between the magnitude and the threshold. After the demodulation, the amplitude is gated into one of two LP digital filters. Filter A averages the previous values of the lower valued amplitude, and Filter B averages the previous values, higher valued amplitudes.

As an example of the operation of this technique, consider a strong signal in sync whose two output magnitudes are distributed about the threshold. After a short time, the threshold will go to the mean of the signals. If the signal frequency then goes into a deep fade, the threshold will diminish and attempt to follow the current mean of the two received amplitudes.

The difference equation which is used to perform the low pass filtering operation may be written:

$$Y(k) = \alpha X(k) + \beta Y(k - 1)$$

where

$$\alpha = 1 - e^{-T'/\tau}$$

$$\beta = e^{-T'/\tau}$$

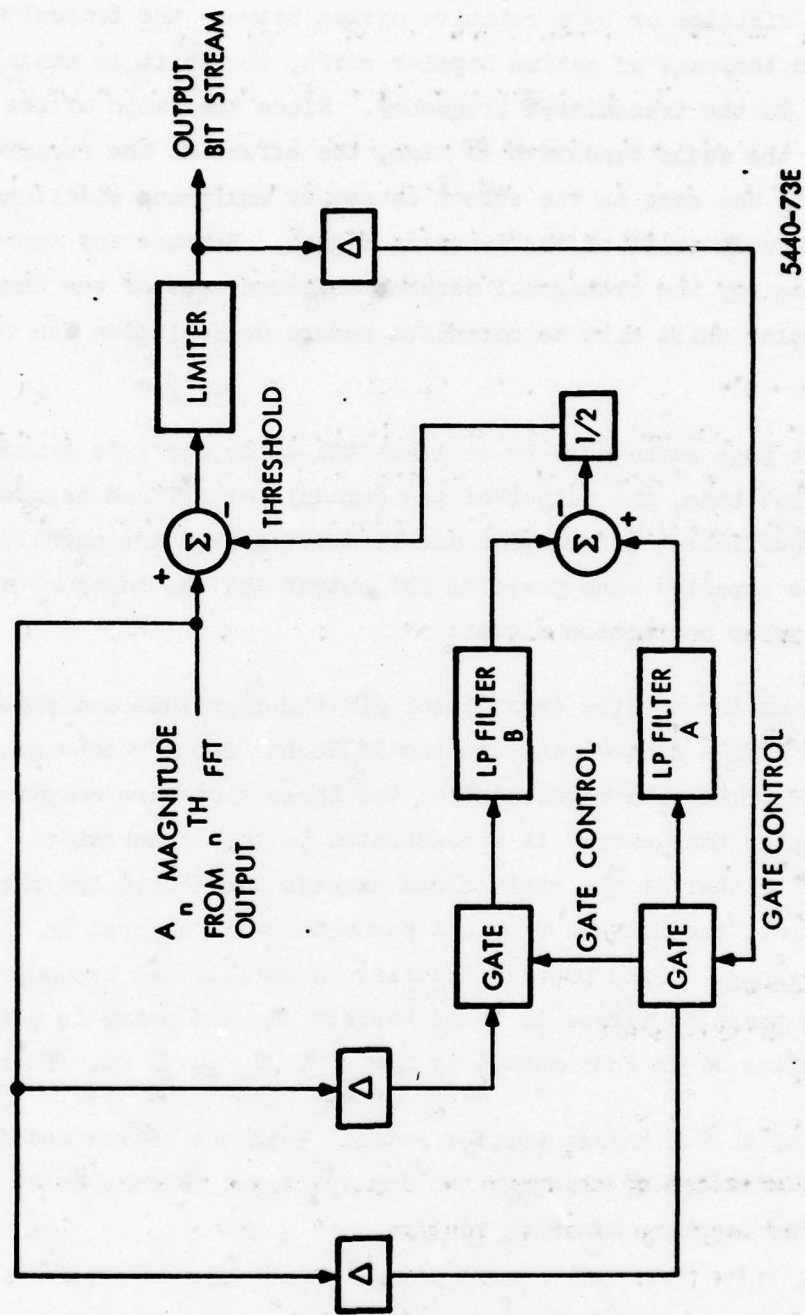
$T'$  = baud period

$\tau$  = filter time constant

$X(k)$  = input

$Y(k)$  = filtered output

This filter has the same impulse response in the sampled time domain as a simple RC integrator with a time constant of  $\tau$ . The normalizing factor,  $\alpha$ , is chosen to give the filter a gain of unity. In the program,  $\alpha$  is the parameter EZ, and  $\beta$  is the parameter AT. The actual value of  $T'/\tau$  used in the program will be 1/5. Thus, the filter essentially takes a weighted average of the previous 25 input values (five time constants).



5440-73E

Figure 3.1-7. Amplitude Demodulator

### 3.1.3 Continuous Doppler Correction

Doppler shift can be caused by a frequency offset in the radio transmitter and receiver circuits or by a relative motion between the transmitter and receiver. In the case of motion Doppler shift, the shift is actually proportional to the transmitted frequency. Since the ratio of the carrier frequency to the audio bandwidth is high, the effect at the received audio is essentially the same as the effect caused by equipment misalignment - a constant frequency shift of the incoming signal. Because any appreciable shift will destroy the orthogonal separation properties of the demodulation FFT, the Doppler shift must be corrected before demodulation can be properly performed.

If a constant tone surrounded by at least two empty slots is transmitted as a Doppler pilot tone, the output of the demodulator FFT can be used to track this tone. Sufficient information can be derived from the magnitudes and/or phases of the expected tone position FFT output and the adjacent outputs to compute a Doppler correction signal.

After each pass through the demodulator FFT the magnitude and phase of the Doppler tone (FFT - slot 4) and the two adjacent "empty" slots are calculated in subroutine CTOP. The magnitudes of the three slots are compared to verify that the Doppler tone energy is concentrated in the center slot. If the magnitudes of either of the empty slots exceeds one-fourth the magnitude of the center slot, the frequency offset parameter  $\omega$  is stepped in the appropriate direction ( $\pm 1$ hz). Fine Doppler tracking is established by comparing the Doppler tone phase on a frame to frame basis.\* If the phase is advancing the offset parameter  $\omega$  is incremented by  $\Delta = \frac{1}{2\pi} \cdot \frac{f_s}{2^{15}} \approx \frac{1}{32}$  hz. This allows tracking of up to 2.3 hz/sec Doppler rates. Both the coarse and fine tracking rates can be adjusted by changing two instructions. Figure 3.1-8 presents a block diagram of the Doppler tracking routing.

\*Corrections for non-orthogonal guard times are made where applicable.

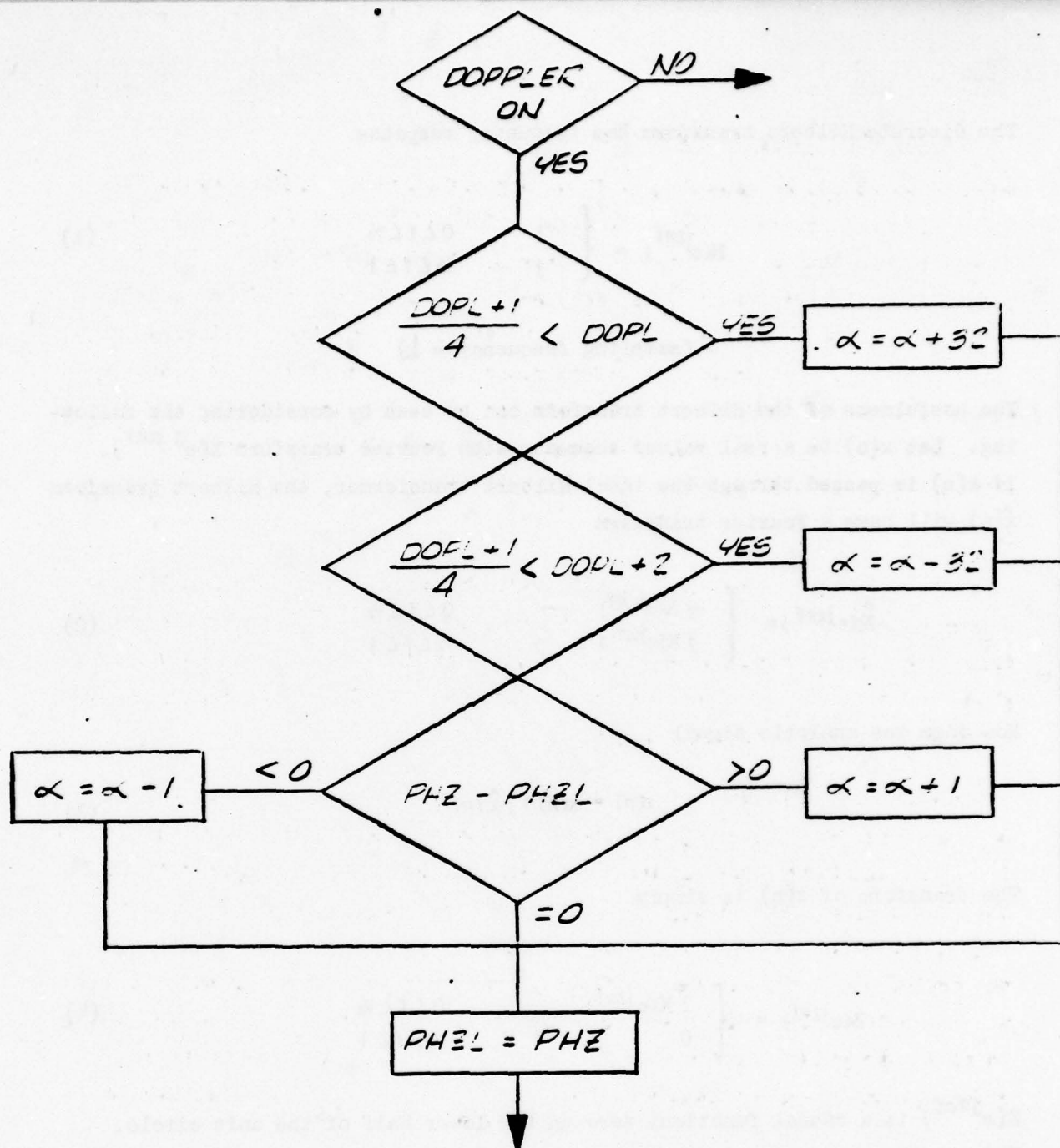


FIGURE 3.1-8  
DOPPLER-TRACKING

The discrete Hilbert transform has frequency response

$$H(e^{j2\pi f}) = \begin{cases} -j & 0 \leq f < \frac{1}{2} \\ j & \frac{1}{2} \leq f < 1 \end{cases} \quad (1)$$

(sampling frequency = 1)

The usefulness of the Hilbert transform can be seen by considering the following. Let  $x(n)$  be a real valued sequence with Fourier transform  $X(e^{j2\pi f})$ . If  $x(n)$  is passed through the ideal Hilbert transformer, the Hilbert transform  $\hat{x}(n)$  will have a Fourier transform

$$\hat{X}(e^{j2\pi f}) = \begin{cases} -j X(e^{j2\pi f}) & 0 \leq f < \frac{1}{2} \\ j X(e^{j2\pi f}) & \frac{1}{2} \leq f < 1 \end{cases} \quad (2)$$

Now form the analytic signal

$$z(n) = x(n) + j \hat{x}(n) \quad (3)$$

The transform of  $z(n)$  is simply

$$Z(e^{j2\pi f}) = \begin{cases} 2 X(e^{j2\pi f}) & 0 \leq f < \frac{1}{2} \\ 0 & \frac{1}{2} \leq f < 1 \end{cases} \quad (4)$$

$Z(e^{j2\pi f})$  is a causal function, zero on the lower half of the unit circle. This is the key to frequency shifting with the Hilbert transform. The procedure is outlined in figure 4.1-9. The analytic signal  $Z(n)$  is multiplied by the complex exponential  $\exp\{j^2 \pi \Delta f n\}$ . This time domain multiplication yields convolution of the causal  $Z(e^{j2\pi f})$  with an impulse at  $f = \Delta f$ , thus effecting a shift of  $\Delta f$  to the spectra. As illustrated in figure 1, the real part of this complex signal, call it  $y(n)$ , is the frequency shifted equivalent of

the real input signal  $x(n)$ . This procedure can be realized with two real multiplications and one real addition (in addition to the operations required to generate  $\hat{x}(n)$  from  $x(n)$  ).

In summary, the frequency shifted signal  $y(n)$  can be expressed

$$y(n) = x(n) \cos [2\pi\Delta f_n] - \hat{x}(n) \sin [2\pi\Delta f_n]$$

If  $\Delta f$  is negative this procedure shifts the spectra toward zero.

The FIR Hilbert transformer used in the modems was designed at NSA and has the impulse response shown in figure 3.1-10. The magnitude and phase responses of this filter are illustrated in figures 3.1-11 and 3.1-12.

Once the Hilbert transform of the input has been calculated the sine and cosine of the offset frequency use required to complete the procedure. An NSA designed resonator is used for this purpose. Figure 3.1-13 presents a block diagram of the resonator. The difference equations that characterize the resonator are:

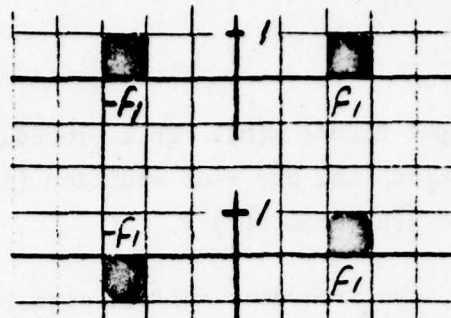
$$\begin{aligned} C(i+1) &= C(i) + \alpha S(i) \\ S(i+1) &= S(i) - \alpha C(i+1) \end{aligned}$$

For initial conditions  $C(0)=1$ ,  $S(0)=0$  the response can be shown to be<sup>1</sup>

$$\begin{aligned} C(n) &= \cos [n\alpha] \\ S(n) &= -\sin [n\alpha] \end{aligned}$$

<sup>1</sup> The NSA Hilbert Transformer and Resonator designs are analyzed in Analysis of the Hilbert Transformer Frequency Shifting Technique, W. Carmichael System Engineering Technical Memorandum No. 16. 1974.

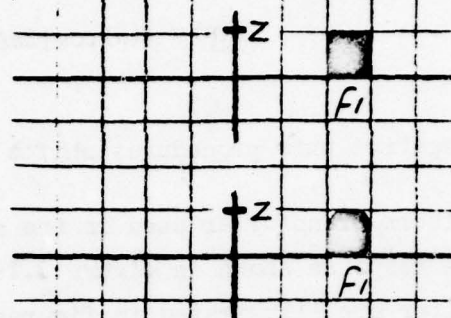
REAL SIGNAL  $x(n)$



REAL

ANALYTIC SIGNAL

$$z(n) = x(n) + j\hat{x}(n)$$

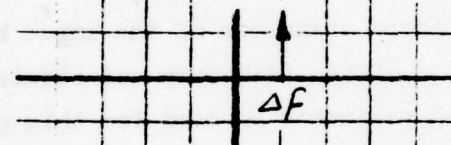


REAL

IMAG

COMPLEX EXPONENTIAL

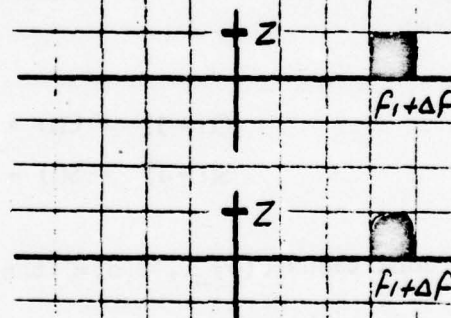
$$e^{j2\pi\Delta f n}$$



REAL

SHIFTED ANALYTIC SIGNAL

$$z(n)e^{j2\pi f_0 n}$$

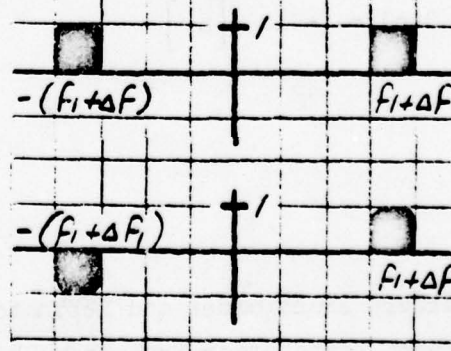


REAL

IMAG

SHIFTED REAL SIGNAL

$$\text{Re}\{z(n)e^{j2\pi f_0 n}\} = y(n)$$

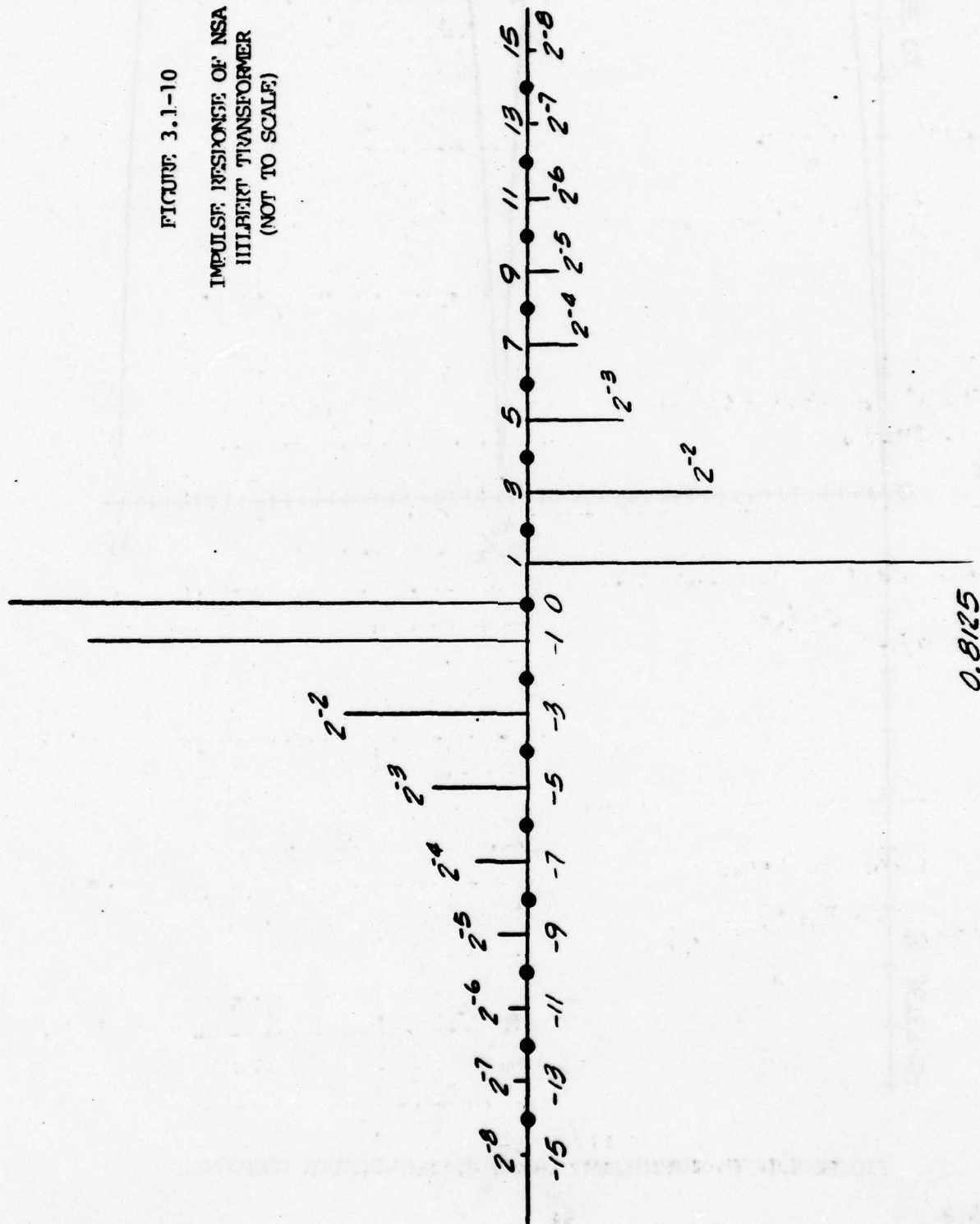


REAL

IMAG

FIGURE 3.1-9  
HILBERT TRANSFORM FREQUENCY SHIFTING PROCEDURE

FIGURE 3.1-10  
IMPULSE RESPONSE OF NSA  
HILBERT TRANSFORMER  
(NOT TO SCALE)



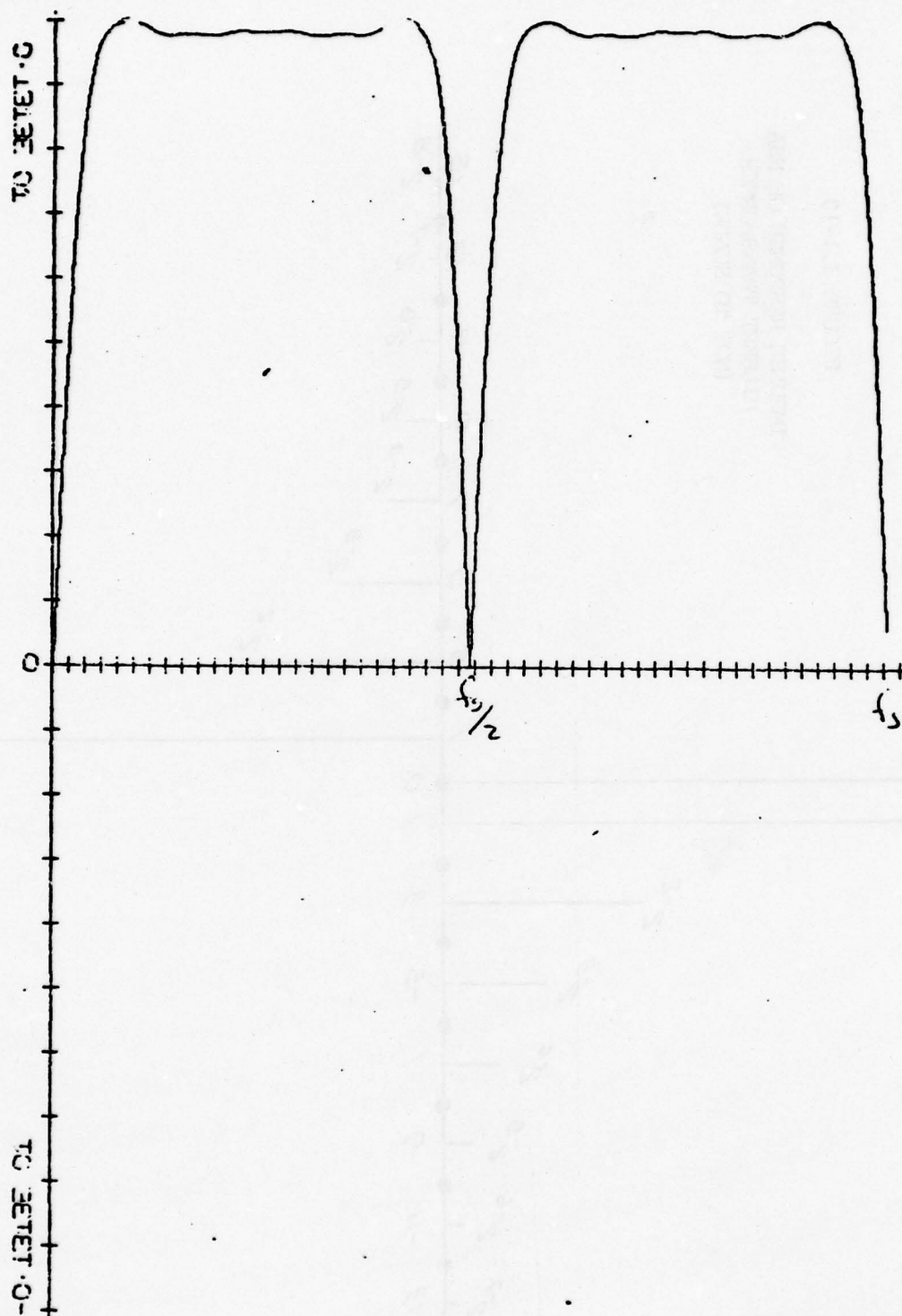


FIGURE 3.1-11 FIR HILBERT TRANSFORMER MAGNITUDE RESPONSE.

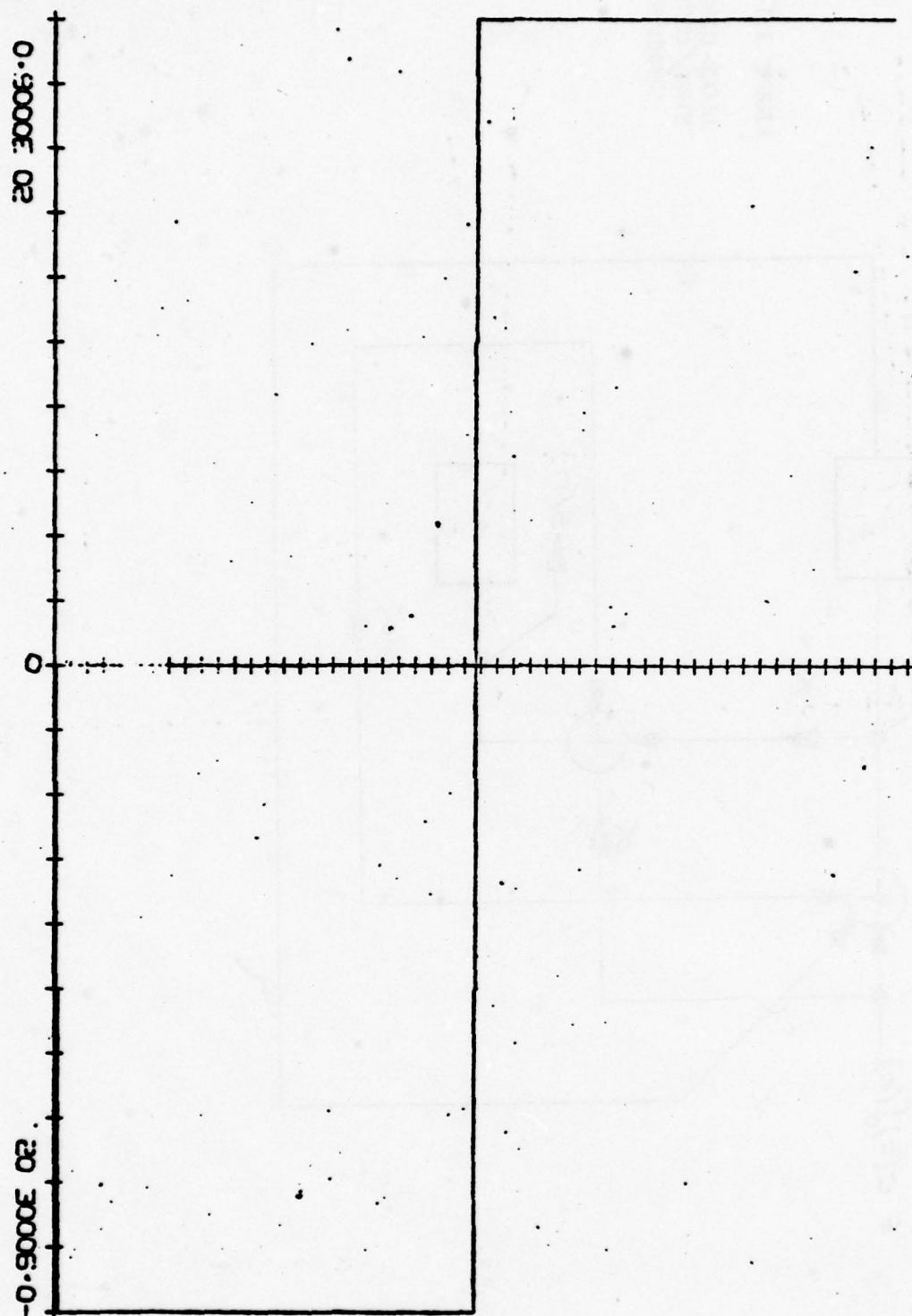


FIGURE 3.1-12 PHASE RESPONSE OF FIR HILBERT TRANSFORMER.

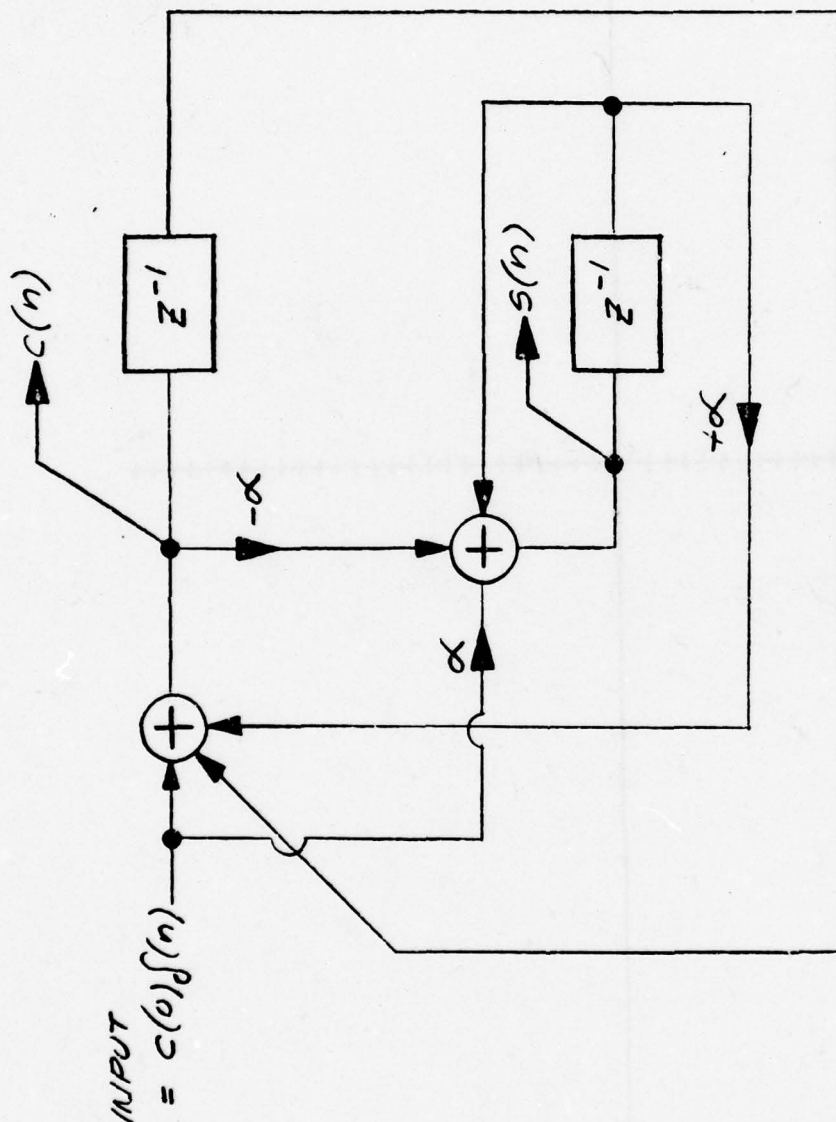


FIGURE 3.1-13  
BLOCK DIAGRAM  
SINE/COSINE  
GENERATOR

$$e^{j\omega_c t_n} = \cos \omega_c t_n - j \sin \omega_c t_n \quad (4-6)$$

where

$\omega_c$  = the output frequency offset

This operation results in a unidirectional frequency shift of the output spectrum. The operation is essentially the discrete version of the Fourier Transform shift theorem. The frequency of  $\omega_c$  is continually updated by the Doppler correction coefficient so as to shift the frequency displaced input back into its proper representation in the frequency domain.

The sinusoids  $\cos \omega_c t_n$  and  $j \sin \omega_c t_n$  are generated by a table look-up algorithm which uses the FFT cosine table. Since the frequencies generated are not necessarily orthogonal to the table, an interpolation algorithm must be used to generate these frequencies.

If the highest tone in the set is too close to 1/2 the sampling rate, this correction algorithm, which is efficient in terms of computation time, cannot be used. The reason is that it shifts both the upper and lower side bands together. In a digital representation, the frequencies in the tone set are reflected about half the sampling rate. The reflected values which are shifted along with the desired frequencies may not be orthogonal to the FFT slot and can cause interference with the measurements if they are too close to the  $\sin x/x$  response of the nearest demodulation FFT slot.

The method of Doppler correction that will be used in most of the modes is to form a set of signals which consist of a set of signals one of which is the Hilbert Transform of the other.

### 3.1.4 CONTINUOUS SYNCHRONIZATION

The algorithm used to process the synchronization input data may be represented by the following difference equation:

$$y_{k+1} - 2(\cos \omega_p T) y_k + y_{k-1} = g_k \quad (2-7)$$

where  $\omega_p$  is the radian center frequency of the filter,  $T$  is the sampling period, and  $g_k$  is the input. This difference equation has the solution

$$y_k = \frac{1}{\sin \omega_p T} \sum_{n=1}^k g(n) \sin (k-n) \omega_p T + C_1 \cos (k \omega_p T) + C_2 \sin (k \omega_p T) \quad (2-8)$$

where the last two terms represent the transient part of the solution and may be dropped by starting the difference equation with zero initial conditions [ $y_0 = 0$  and  $y_1 = 0$ ]. If this difference equation is excited by an input waveform until time  $(K' - 1)$  and then if the difference equation is allowed to ring (i.e.,  $g_k = 0$ ,  $k \geq K'$ ), the output waveform for  $k \geq K'$  will be a sinusoid with a frequency of  $\omega_p$  whose amplitude and phase will be related to the input amplitude, phase, and frequency.

Figure 3.1-14 shows the response to an input sinusoid

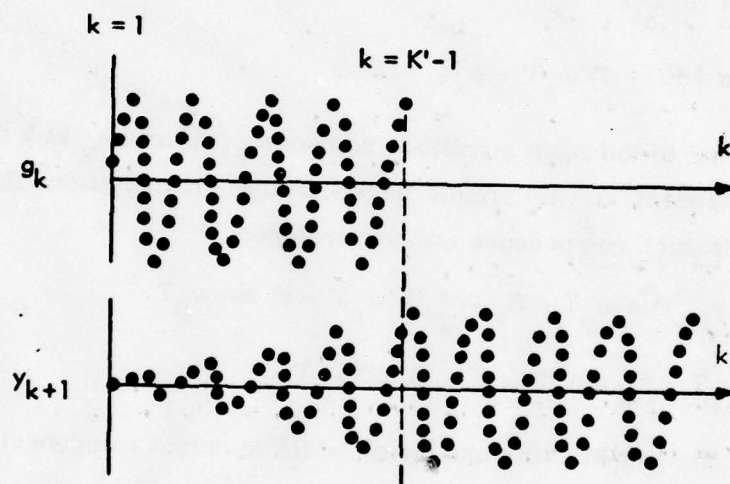
$$g_k = A \cos (\omega_p k T + \phi) \quad (2-9)$$

whose frequency is identical to the center frequency of the difference equation. The output of the filter for  $kT \geq K'T$  can be written from equation (2-8) as:

$$y_k = \frac{A}{\sin \omega_p T} \sum_{n=1}^k \frac{1}{2} [\cos (\phi + k \omega_p T) - \cos (2n \omega_p T - \phi - k \omega_p T)]$$

for large  $k$

$$y_k \sim - \frac{A}{\sin \omega_p T} \left[ \frac{k}{2} \cos (\phi + k \omega_p T) \right] \quad (2-10)$$



1254-70E

Figure 3.1-14. Response of Difference Equation to an Input Sinusoid

Thus, when the filter is excited at its center frequency, the envelope amplitude is directly proportional to the input amplitude and the number of input samples processed. The phase of the output is the same as the input phase.

In the synchronization process, 127 or 63 values of input are processed and the last input sample is set to zero to allow the difference equation to ring once. The last two values of output ( $y_{K'}$  and  $y_{K'+1}$  where  $K'$  is 128 or 64) are used to calculate the quadrature components which are in turn used to compute the magnitude.

The algorithm used to calculate the quadrature components is derived by considering the output points as two samples of a sinusoid:

$$y_{K'} = A_x \sin (K' \omega_p T + \phi_x) \quad (2-11)$$

$$y_{K'+1} = A_x \sin [(K' + 1) \omega_p T + \phi_x]$$

where  $A_x$  is proportional to the input amplitude component,  $A$ , and  $\phi_x$  is a function of the phase of the input component,  $\phi$ . By simple trigonometric manipulation, the following equations for the quadrature components can be developed:

$$R_p = y_{K'+1} - y_{K'} \cos \omega_p T = A_x \cos (K' \omega_p T + \phi) \sin \omega_p T \quad (2-12)$$

$$I_p = y_{K'} \sin \omega_p T = A_x \sin (K' \omega_p T + \phi) \sin \omega_p T \quad (2-13)$$

Given the quadrature components, the magnitude of filter output component can be calculated. The algorithm employed to find the approximate square root of the sum of the squares is described by the following expression.

$$m = \alpha Q + \beta P \sim \sqrt{R_p^2 + I_p^2} \quad (2-14)$$

where

$m$  = calculated magnitude

$\alpha = 1$

$\beta = 1/2$

$Q = \text{Max } R_p, I_p$

$P = \text{Min } R_p, I_p$

The rms percentage error incurred as a result of using this approximation is 7.21 percent.<sup>4</sup> The mean squared error could be lowered by making  $\beta$  equal to  $1/4$ , but the resultant peak error would be increased.

The combination of difference equation and magnitude algorithm has a  $\sin x/x$  frequency response centered at  $\omega_p$ . The first zero crossings occur at the reciprocal of the integration period away from the center frequency. The integration time of the filter is made to equal the orthogonal period of the tone set and  $\omega_p$  is chosen to be harmonically related to the transmitted tones so that the zero crossings of the difference equation response fall on the center frequencies of the information bearing tones.

In slot synchronization, the difference equation is centered on an orthogonal frequency at which no energy is transmitted. Since there is no tone transmitted in the synchronization slot, the presence of energy at this frequency is caused by a baud transition which spills energy from the transmitted tones into the slot. If there is no baud transition in the measurement interval, the value of  $m$  will go to zero because of the orthogonal relationship between the transmitted tones and the zero crossings of the response.

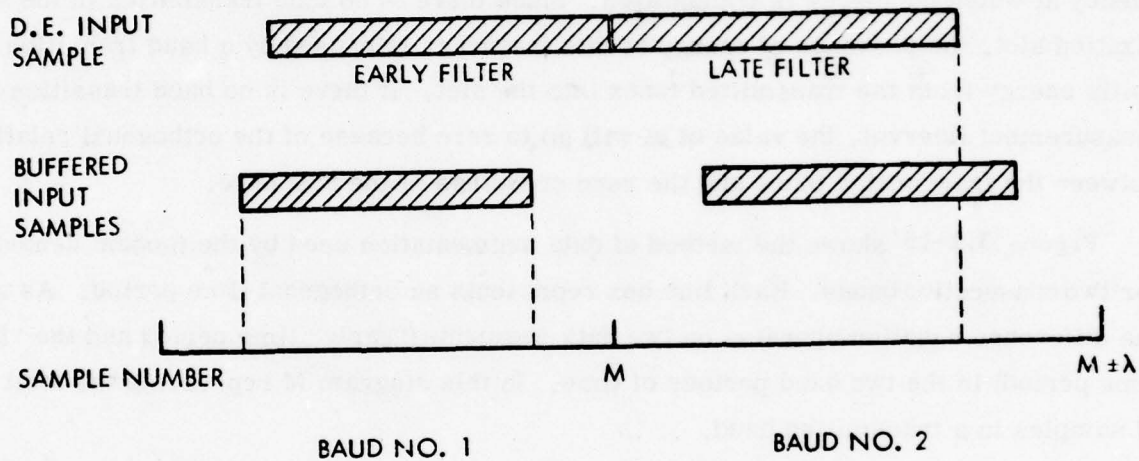
Figure 3.1-15 shows the method of data segmentation used by the modem demodulator for two consecutive bauds. Each line box represents an orthogonal time period. As shown, the difference equation operates on two data segments ("early" time period and the "late" time period) in the two baud periods of time. In this diagram  $M$  represents the total number of samples in a transmitted baud.

Baud synchronization is accomplished by adding a correction,  $\pm \lambda$ , to the number of samples processed every other baud. The difference equation filter can detect four alignments of the actual received baud with respect to receiver timing. These conditions are diagrammed in Figure 3.1-16.

In cases "a" and "b" of Figure 3.1-16 energy is measured in only one of the two intervals, the synchronization correction is calculated from the following expressions:

$$\lambda = C(m_\ell - m_e) \quad (2-15)$$

4. Manley, H.J., "A Simple Approximation to  $\sqrt{X^2 + Y^2}$  with Minimum Mean Squared Percentage Error," ARL Research Note 733, February 1960.



1241-70E

Figure 3.1-15: Data Input Timing Diagram

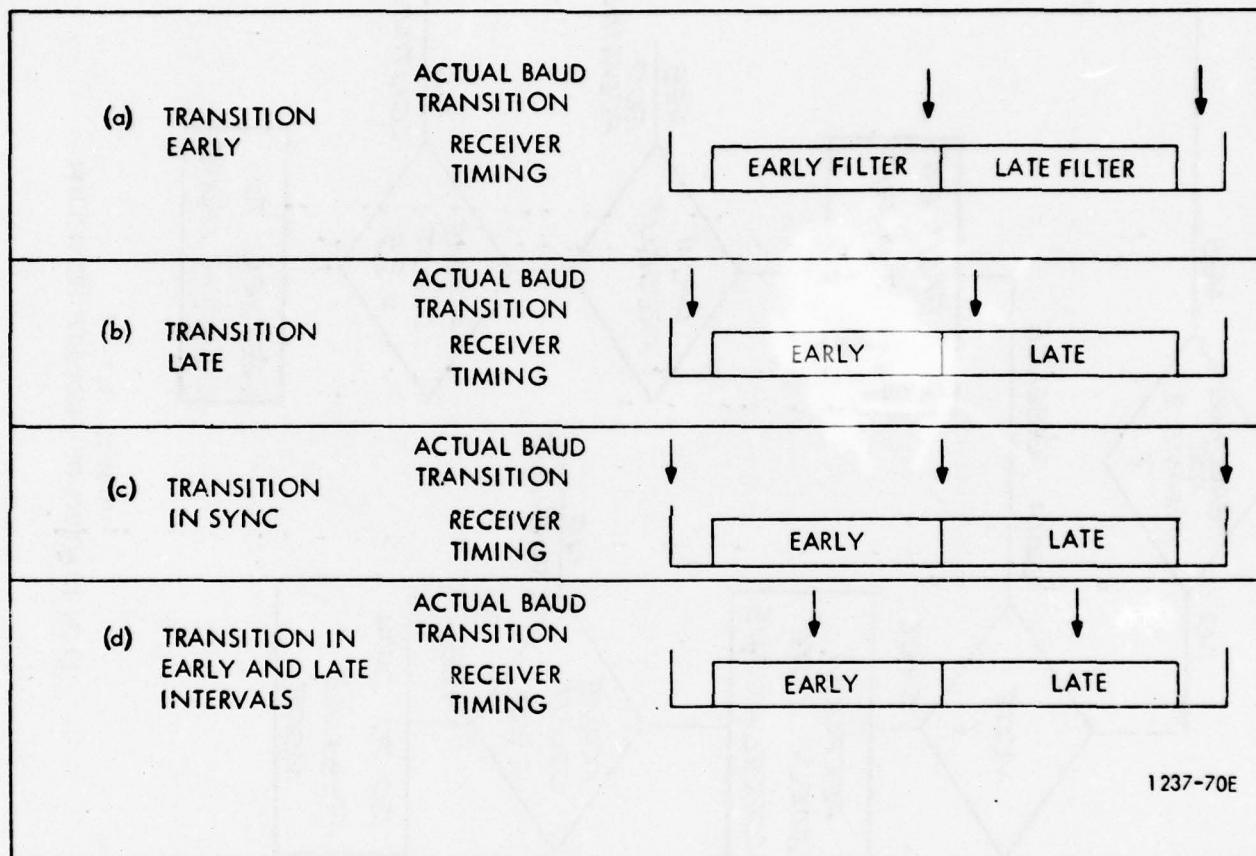


Figure 3.1-16: Synchronization Alignment

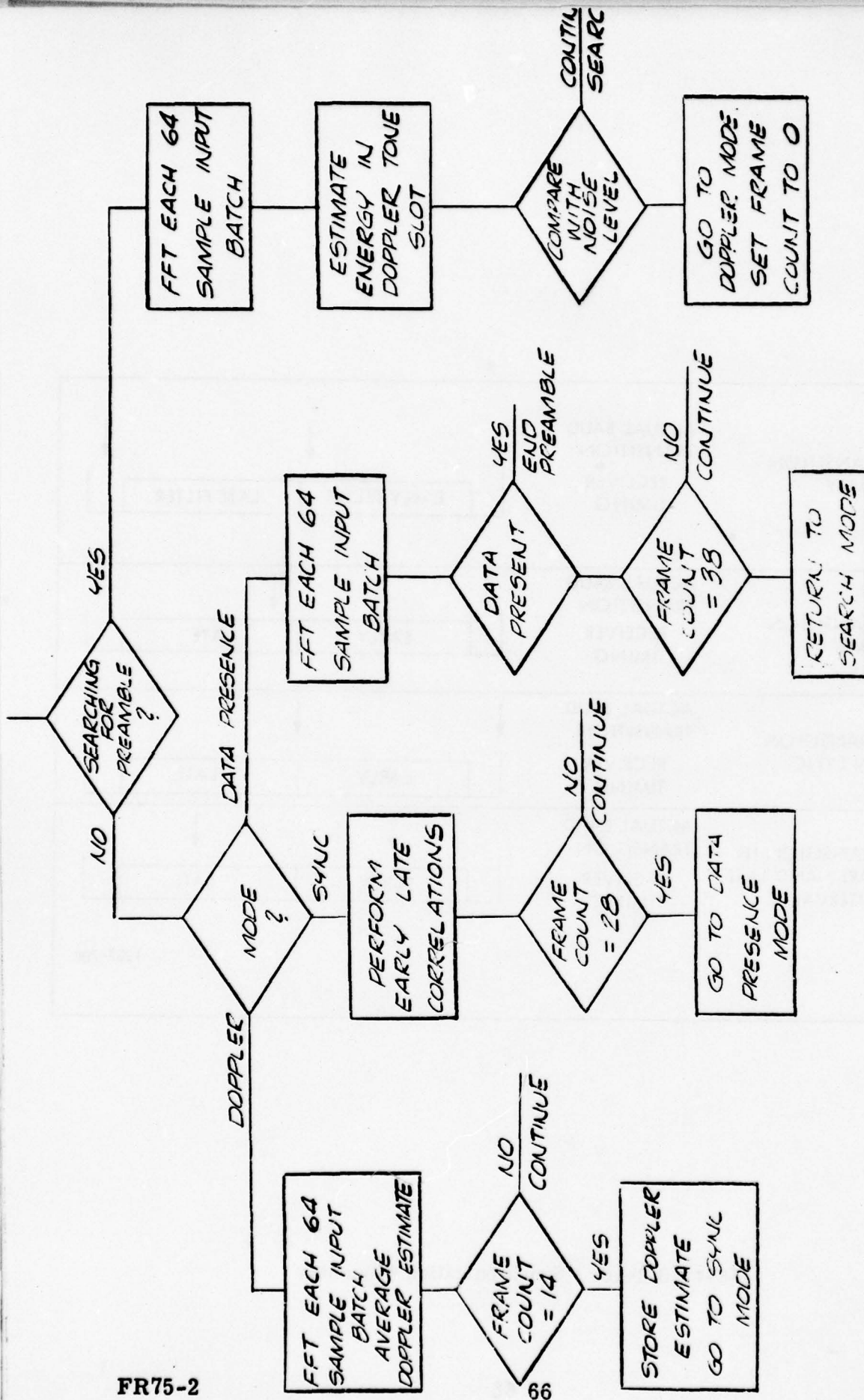


FIGURE 3.1-17  
FLOW DIAGRAM OF PREAMBLE DETECTION

where

$m_l$  = magnitude of late filter  
 $m_e$  = magnitude of early filter  
 $C$  = synchronization constant

The constant,  $C$ , is chosen so that the maximum value of  $\lambda$  during the lock-in phase synchronization is three or four samples.

When the received baud is in sync, neither filter will measure energy in the synchronization slot and the correction goes to zero.

Since the number of samples monitored by the early and late filters exceeds the number of samples in a baud, it is possible that a baud transition will lie in both the early and later filters.

When a preamble time acquisition is used and if the HF link is poor this condition will occur. To prevent ambiguities in such an occurrence, the values of the early and late magnitudes are checked before they are used in equation (2-15). If both magnitudes exceed an experimentally determined threshold, the value of  $\lambda$  is forced to plus one. In this manner, the program will tend to walk out of the indeterminate position.

When the input level is reduced to zero, as might be experienced in a deep signal fade, the correction goes to zero. Thus, the synchronization will be maintained in a deep fade as long as the accuracy of the transmitter and receiver clocks is sufficient. The continuous synchronization rate will be adjusted so that the timing pulses for the buffered output data will not be more than three microseconds in any one output bit period.

### 3.1.5 PREAMBLE FUNCTIONS

The transmitted preamble duration is 32 frames, consisting of a CW Doppler tone at frequency  $\frac{f_s}{16}$ , and a biphase modulated sync tone at frequency  $\frac{f_s}{8}$ , modulated at the frame rate.

The receive preamble processing performs four functions

- a.) detect preamble presence
- b.) estimate the doppler frequency
- c.) establish frame synchronization
- d.) detect data presence (or end of preamble)

As implemented these four functions are performed sequentially. The block diagram in Figure 3.1-17 summarizes how the preamble function is performed.

Initially an FFT is performed on each block of 64 input samples. The sum of the magnitudes of slots 4, 5 and 6 (doppler shots) are compared with the sum of the slots 18, 19, and 20 (noise slots). If their difference exceeds a relative threshold, the frame counter is set to zero and the doppler estimation mode initialized.

In the doppler estimate mode FFT's are again performed on each 64 sample input block. An average is maintained of dot product values of the latest doppler slot value with the previous doppler slot value. After 14 frames the average is converted to polar form with the angle being the doppler estimate.

In the synchronization mode 16 sample DFT's are performed on "early" and "late" windows. The most negative correlation of the early-late windows corresponds to frame transition. This function is performed for 14 frames. This function requires 370 usec of each two sample period (. 400 usec) thus prohibiting parallel doppler and sync modes.

With the doppler and synchronization modes complete the preamble waits for the first data frame before relinquishing control to the main program. The synchronization and doppler acquisition algorithms are discussed below.

### 3.1.5.1 Synchronization Acquisition

Time synchronization is performed in two parts: acquisition and tracking. Acquisition is performed by detecting the positions of the phase transitions to the sync tone of the received preamble. A discrete Fourier transform is performed, calculating the spectral coefficient on the sync tones. The integrations are performed on successive sequences of sixteen input signal samples, as shown in Figure 3.1-18. If the real and imaginary coefficients of the early and late integrations are designated as  $R_E$ ,  $X_E$ ,  $R_L$ , and  $X_L$  respectively, then the dot product is calculated as

$$E \cdot L = R_E R_L + X_E X_L$$

The locus of the dot product amplitude as a function of the location of the data transition in the integration pair is shown in Figure 3.1-18. When no phase transition occurs during the early or late integrations, the phases of the early and late coefficient vectors are the same, resulting in a maximum positive dot product. When the 180 degrees phase transition occurs at the early-late transition, the resulting dot product is maximum negative. For intermediate transition locations, the dot product locus follows a straight line between the positive and negative maximum.

The dot product is calculated at the end of each integration pair. The criterion for locating the transmitted phase transition is a selection of the integration pair yielding the most negative dot product. The early-late transition of this integration pair is the best estimate of the location of the phase transition. In order to attain the desired resolution, integrations are time staggered as shown in Figure 3.1-19. As shown, 16 sample integrations are staggered in two sample offsets, requiring 8 parallel integrations. This results in a dot product calculation on alternate samples, or a  $\pm 1$  sample resolution.

This detection process is continued over the duration of the transmitted preamble. The location of the most negative dot product is stored, on a continuing basis. Thus, although there is a negative peak for each frame transition, only the position of the most negative peak is retained at the end of the preamble. Since a fade on the sync tone causes reduction in the magnitude of the dot product, non-faded detections are favored over faded ones in a preamble which bridges a fade.

At the end of the preamble the time location of the most negative dot product is noted. A frame transition will have occurred 16 sample time intervals prior to this time, and the receiver frame counter zeroed to the transition frame.

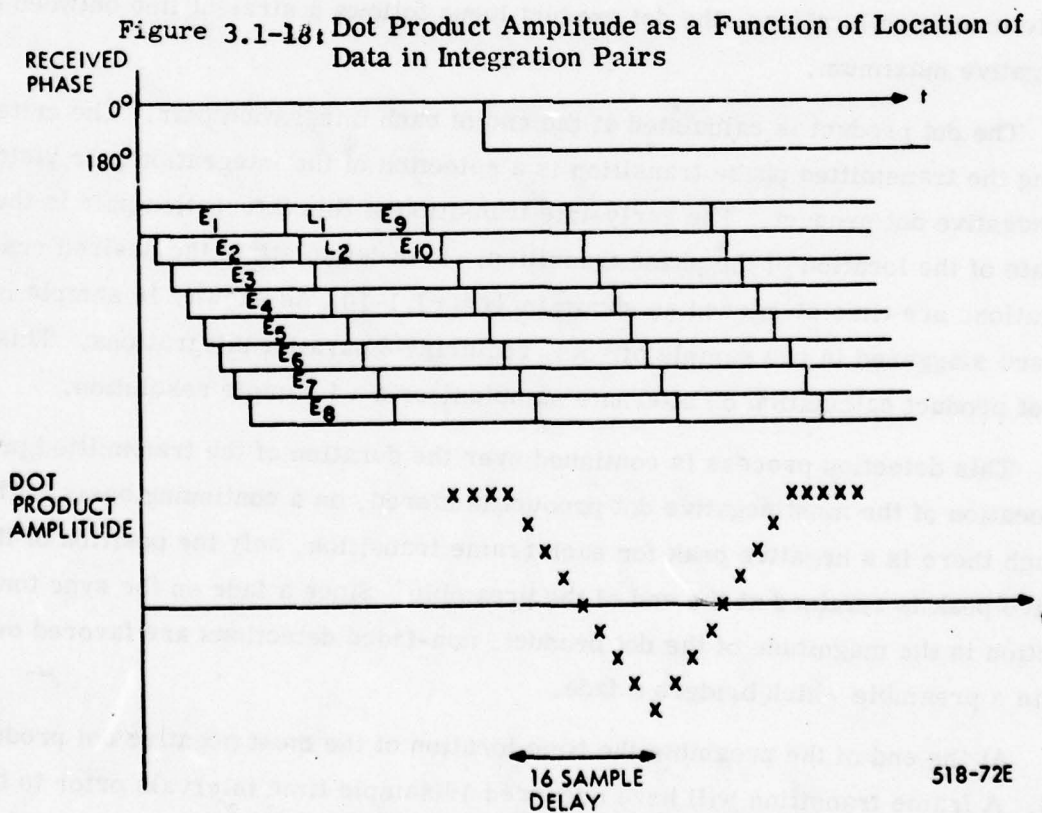
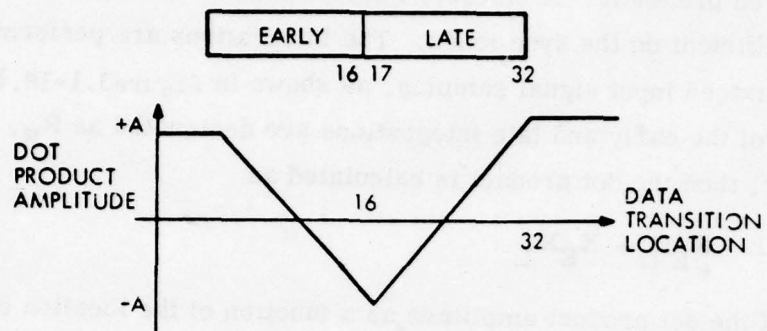


Figure 3.1-19: Preamble Sync Detection

### 3.1.5.2 Doppler Acquisition

In orthogonal signaling schemes, it is necessary to remove frequency errors on the incoming signal prior to orthogonal detection, or crosstalk will result. These errors are of two types: frequency offset due to equipment drift or terminal motion, and Doppler spread resulting from ionospheric scatterer relative motion or terminal motion during multipath transmission. If a density function of the instantaneous frequency error is formed, then frequency offset is the mean, and Doppler spread is the variance. Doppler correction circuits detect and correct the mean frequency error, but cannot eliminate the Doppler spread term.

The initial frequency correction is made during the preamble. Successive 64-sample DFT's are performed to calculate the coefficients of the preamble Doppler tone frequency. Coefficients from successive integrations are vector multiplied to calculate the differential vector, as is done in the PSK data demodulation process. In the case of Doppler detection, however, it is the angle of the differential vector that is desired, as this angle is directly proportional to the frequency offset on the received preamble Doppler tone. This angle is calculated for 64-sample block, and the results are averaged over the preamble detection period for process gain. At the end of the preamble, the final offset calculation is used to perform a step frequency correction on the incoming signal. The initial correction is limited  $\pm f_s/2N$  where  $f_s/N$  is the orthogonal tone spacing. For example the initial offset correction range for the TDP3K-2400 moden is  $\pm 55$  Hz.

### 3.2 CODING TECHNIQUES

The product codes and the "Rank" decoding algorithm used in this application were reported recently in the literature to be the error control technique employed with an experimental error controlled HF modem.<sup>5</sup> Called Codem I, it is designed primarily to provide protection against frequency selective fading, and indeed bit error rate improvements of 1 to 2 orders of magnitude have been observed.

#### 3.2.1 PRODUCT CODES - RANK DECODING

The (25, 16) product code was employed in Codem I with a decoding algorithm suggested by Chase<sup>5</sup>. The decoder makes soft decisions on the 25-bit received word and ranks the relative reliability of each bit. Decoding then proceeds only with the ranking information. The (25, 16) code is similar in structure to the (9, 4) product code which is more convenient to use to describe the Rank decoder. We will follow the example given by Chase closely. To encode a (9, 4) product code word, consider arranging the information bits ( $I_1 - I_4$ ) in a  $2 \times 2$  array, in Figure 4.2-1. The parity bits ( $P_1 - P_5$ ) are formed by taking mod -2 sums of the rows and columns; that is:  $P_1 = I_1 \oplus I_2$ ,  $P_3 = I_1 \oplus I_3$ , and so on.  $P_5$  is formed by summing all four information bits ( $P_5 = I_1 \oplus I_2 \oplus I_3 \oplus I_4 = P_1 \oplus P_2 = P_3 \oplus P_4$ ).

$I_1$	$I_2$	$P_1$
$I_3$	$I_4$	$P_2$
$P_3$	$P_4$	$P_5$

Figure 3.2-1: The (9, 4) Product Code

5. Chase, D., "A Combined Coding and Modulation Approach for Communication over Dispersive Channels", IEEE Trans. on Communication, Vol. Com. 21, March 1973.

The nine bits to be decoded are ranked according to their relative reliability. The bits are labeled 1,2,....., 9 where 9 indicates the most reliable received bit and 1 the least reliable. Consider the possible ranking situation shown in Figure 4.2-2(a). The x's indicate the locations of three assumed errors (the most that a soft decision decoder can find and correct in this case) and the f's indicate flags for the d-1 least reliable bits (d=4 is the (9,4) code's minimum distance).

The decoder begins by checking the parity of each row and column of the received word in Figure 3.2-2(a). A ✓ indicates that the parity checks and ✗ indicates not. Note that a parity equation will check if there are no errors or two errors in a row or column. The rest of the algorithm can be stated in three main parts:

- 1a. Consider the highest ranked unflagged undecoded bit. If all parity equations check, decode the bit as is.
- 1b. If all equations do not check, flag the bit.
- 1c. If at least one equation checks, decode the bit as is, provided its rank is larger than the lowest ranked undecoded bit in each unchecked equation. Flag the lowest ranked undecoded bit in each unchecked equation.
2. If, after decoding a bit, there remains only one undecoded bit in any equation, decode the remaining bit by forcing parity to check.
3. When all remaining bits are flagged, decode the highest ranked flagged bit and go to step 2.

Figures 3.2-2(b),(c), (d), (e) and (f) trace the decoding steps for the previously stated example. We begin with bit 9 and notice that the vertical parity checks but the horizontal parity does not. Invoking Step 1c, we decode bit 9 and circle it to so indicate. Bits 7 and 8 are similarly decoded by Step 1c. As soon as bit 7 is decoded, though, the first column and third row have just one remaining undecoded bit. Step 2 insists that we now decode bits 3 and 4 by forcing the parity in the first column and third row to check. Thus, in Figure 3.2-2(c) bits 3 and 4 are decoded and circled. Bit 4 is circled twice to indicate that it had to be complemented for the third row parity to check and indeed we have corrected our first error in so doing. The highest ranked undecoded bit (bit 6) is decoded next by Step 1c, as shown in Figure 3.2-2(e) leaving bit 5 the only undecoded bit in the second row. Step 2 again insists that bit 5 be complemented and decoded next to assure correct parity on the second row. We have just found and corrected the second error (Figure 3.2-2(e)). Finally, bit 2 is decoded by Step 2 enabling bit 1 to also be decoded properly by Step 2, correcting the third error and completing the decoding process.

a.	9	$f^1x$	$f^2x$
	$f^3$	$5x$	$6x$
	7	8	$4x$ $x$
	✓	✓	

b.	9	$f^1x$	$f^2x$
	$f^3$	$5x$	$6x$
	7	8	$f^4x$ $x$
	✓	✓	

c.	9	$f^1x$	$f^2x$
	3	$5x$	$6x$
	7	8	$4$ $x$
	✓	✓	✓

d.	9	$f^1x$	$f^2x$
	3	$f^5x$	$6x$
	7	8	$4$ $x$
	✓	✓	✓

e.	9	$f^1x$	$f^2x$
	3	$5$	$6$ ✓
	7	8	$4$ ✓ $x$
	✓	$x$	✓

f.	9	$1$	$2$ ✓
	3	$5$	$6$ ✓
	7	8	$4$ ✓ $x$
	✓	✓	✓

5448-73E

Figure 3.2.2: Decoding a (9, 4) Product Code

### 3.3 Deliverable Modem Software

This section describes the modem software that was developed for the contract. Four basic programs were written to perform the seven full and half-duplex modem configurations.

Each of the four basic deliverable programs is discussed separately in subsections 3.3.1 to 3.3.4. The parameters of all the modes (Tables 3.3-1 3.3-2 3.3-3 3.3-4), a block diagram of the processing, and memory requirements for the four programs are presented in each section. A timing analysis is also presented for each mode.

Programs were combined to make functional processing simple where common algorithms are used for similar programs and also to take advantage of existing software routines. The four basic programs are briefly described below.

### 3.3.1 2400 Bit Per Second TDPSK, No Coding

Table 3.3-1 shows the parameters of the 2400 b/s uncoded TDPSK program,

This 16-tone, duplex DPSK modem was designed to be compatible with the specifications set forth in "Modem, Digital Data MD-5B( )/G" for the 2400-b/s mode. The program has been interfaced with an AN/ACQ-5 modem operating in the voiced mode.

Figure 3.3-1 is a block diagram of the modem. The input to the modulator is either a fixed 32-bit input frame

or a random data input. At the baud rate (75 Hz) 16 differential phase shift angles are formed - one for each of the sixteen information bearing tones. Each differential phase shift is one of four possible angles. Thus two input data bits are encoded in each information bearing tone.

The table look-up modulator forms 17 sinusoidal output tones from an 128-point cosine table, and outputs the composite signal through a D/A converter and low pass filter. Each sample of the output is formed at the 7040-Hz sampling rate in the interrupt routine.

Figure 3.3-2 (a) shows the amplitude versus frequency plot of the output waveform. The 605-Hz Doppler correction tone is continuous and conveys no data information.

The table look-up modulator generates the output tones by stepping through the cosine table at various rates. For instance, the Doppler tone is generated by outputting every eleventh value (mod 128) in the table at the sampling rate. Phase modulation is accomplished by adding increments which correspond to a 45, 135, 225 or 315 degree phase shift to the storage locations which hold the current cosine table address for each of the sixteen information bearing tones. This phase modulation operation is performed at the end of each baud (once each 94 output samples).

TABLE 3.3-UNCODED TDPSK MODEM PARAMETERS

	<u>2400-b/s Mode</u>	<u>3600-b/s Mode</u>	<u>4800-b/s Mode</u>
<u>DATA</u>			
Information Rate	2400 b/s	3600 b/s	5000 b/s
Frame Rate	75 Baud/s	75 Baud/s	66.66 Baud/s
Number Inf. Bits/Frame	32	48	72
<u>SIGNALLING</u>			
Modulation	4 $\phi$ TDPSK	4 $\phi$ TDPSK	8 $\phi$ TDPSK
Number of Inf. Tones	16	24	24
Tone Spacing	110 Hz	93.75 Hz	93.75 Hz
Baud Period	13.33 ms	13.33 ms	15 ms
Orthogonal Period	9.09 ms	10.66 ms	10.66 ms
Guard Time	4.24 ms	2.66 ms	4.33 ms
Doppler Tone	605 Hz	468.75 Hz	468.75 Hz
Information Tones	935-2585 Hz	656.25-2812.5 Hz	656.25-2812.5 Hz
<u>PROCESSING</u>			
Sampling Rate	7040 Hz	6000 Hz	6000 Hz
Samples/Baud	94	80	90
Modulation DFT	Table Look-up	64-Point Complex FFT	64-Point Complex FFT
Demodulation DFT	64-Point Complex FFT	64-Point Complex FFT	64-Point Complex FFT
Doppler Correction	Complex Shift	Hilbert Xform	Hilbert Xform
Synchronization	Slot	Slot	Slot

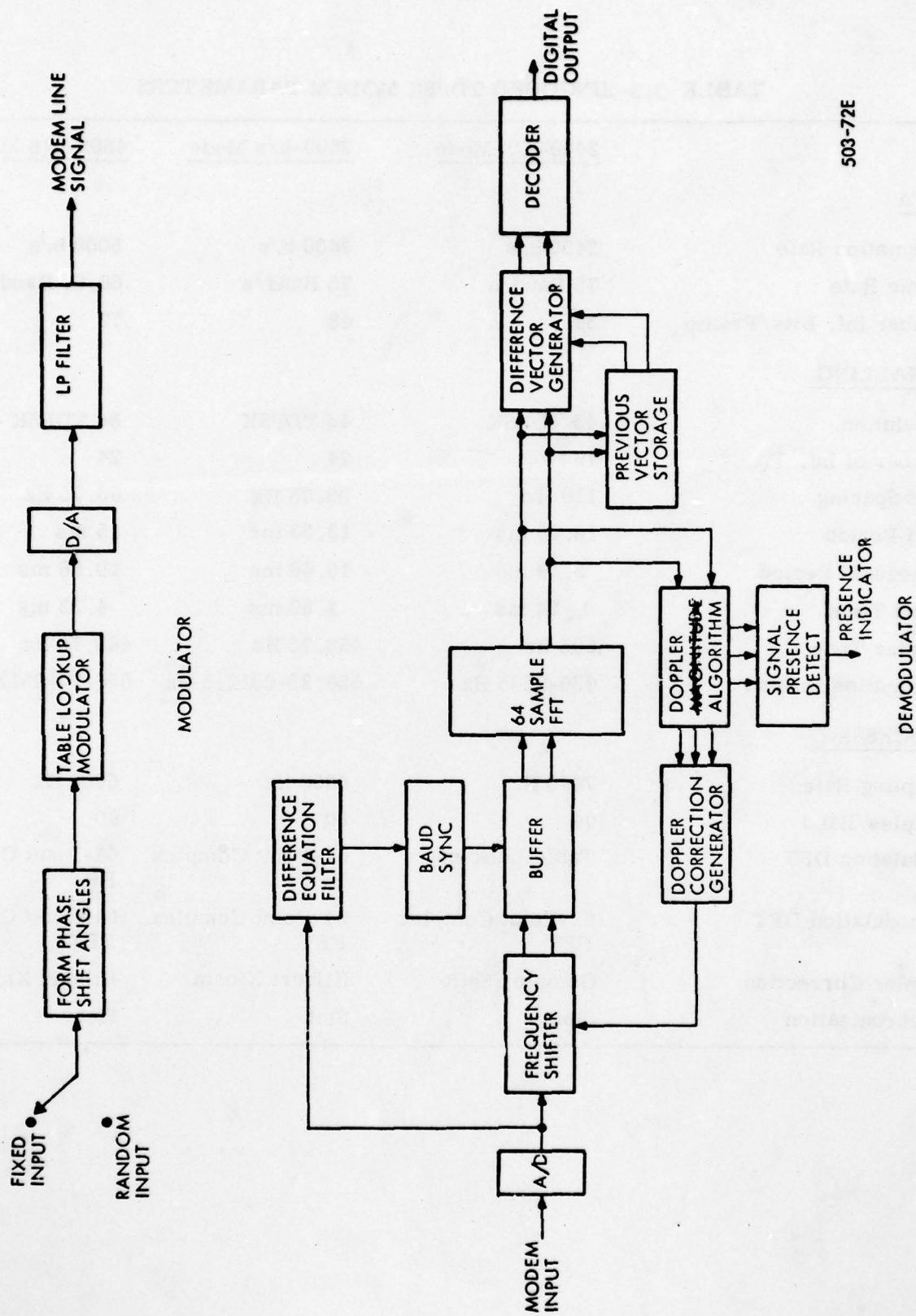


Figure 3.3-1: TDPSK 1/2 Duplex Modem, 2400 b/s Mode

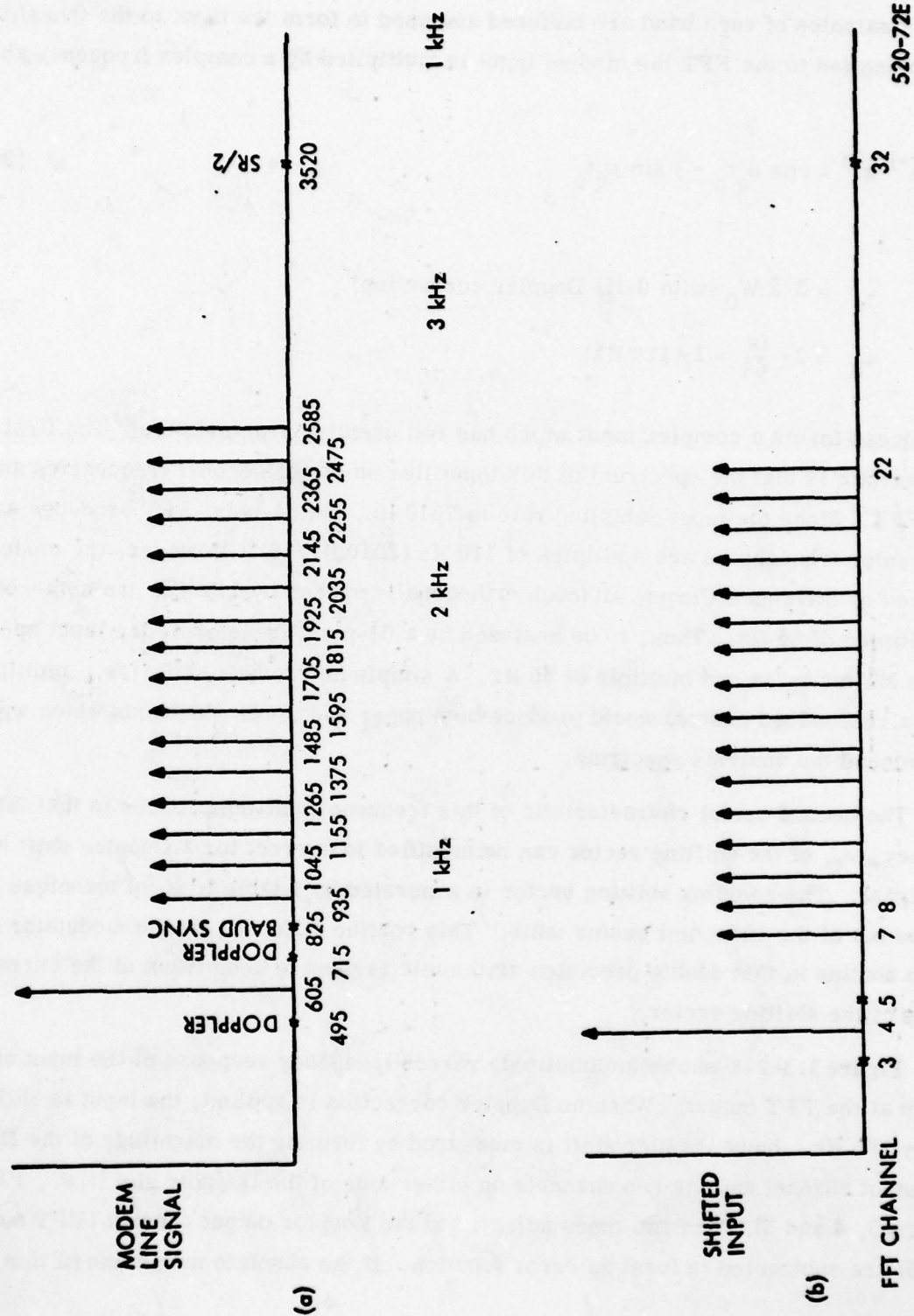


Figure 3.3-2: Amplitude Versus Frequency Plots - Output Waveform and Input at FFT Output

Demodulation analysis is accomplished in a radix-4, 64-point complex FFT. The center 64 samples of each baud are buffered and used to form the input to the transform. Before being fed to the FFT the modem input is multiplied by a complex frequency shifting vector:

$$e^{-j\omega_q t_n} = \cos \omega_q t_n - j \sin \omega_q t_n \quad (2-16)$$

where

$$\omega_q = 3/2 W_0 \text{ (with 0-Hz Doppler correction)}$$

$$\omega_0 = 2\pi \frac{f_s}{64} = 2\pi(110 \text{ Hz})$$

This process forms a complex input which has two useful characteristics. The first such characteristic is that the spectrum of this input lies on the orthogonal frequencies analyzed by the FFT. Since the input sampling rate is 7040 Hz, the 64-point FFT produces a spectrum whose center frequencies are multiples of 110 Hz (3040 Hz/46). However, the modem tones as defined by the Signal Corps, although orthogonally spaced by 110 Hz, are centered on odd multiples of 55 Hz. Thus, to be analyzed by a 64-point transform, the input spectrum must be shifted by an odd multiple of 55 Hz. A simple heterodyne shift (i.e., multiplication by a real valued cosine) would produce both upper and lower sidebands which would in turn confound the analysis spectrum.

The second useful characteristic of this frequency shifting process is that the center frequency,  $\omega_q$ , of the shifting vector can be modified to correct for a Doppler shift in the input signal. The complex shifting vector is generated by a table look-up technique which operates out of the 128-point cosine table. This routine differs from the modulator table look-up routine in that double precision arithmetic is used to keep track of the current table address of the shifting vector.

Figure 3.3-2 (b) shows an amplitude versus frequency response of the input as it appears at the FFT output. When no Doppler correction is applied, the input is shifted down by 165 Hz. Input Doppler shift is measured by forming the magnitude of the Doppler FFT output channel and the two channels on either side of the Doppler slot (i.e., FFT output numbers 3, 4 and 5). The two tones adjacent to the Doppler output channel (FFT numbers 3 and 5) are subtracted to form an error function. If the absolute magnitude of this function

exceeds a threshold, a Doppler correction will be applied. The sign of the function determines the direction of the correction. If a Doppler correction is to be made, a second function is calculated by subtracting the magnitude at the Doppler position (FFT slot number 4) from the adjacent magnitude in the direction of the previously determined Doppler shift. This second function is used to provide a fast or slow update (depending on the sign of this function) to the complex shifting vector.

The sinusoids required to generate the shifting vector are computed in a table look-up routine which uses the 128-sample cosine table. Since the frequencies of these sinusoids are not necessarily harmonics of 55 Hz (the fundamental generated by outputting the table at the sampling rate) an interpolation technique is used in the table look-up routine. The interpolation technique uses double precision arithmetic to define a fractional place in the table for each table look-up operation. The fraction is in turn rounded off for each table look-up operation. This operation is essentially equivalent to a zero order hold interpolation function.

Synchronization is accomplished by using a single pole difference equation centered at 825 Hz to measure baud transitions. In the modem modulator, no tone is transmitted in this slot. Since the difference equation is dumped at an orthogonal interval, any energy measured in the interval will be due to a baud transition. The difference equation is turned on once per baud and aligned to measure adjacent "early" and "late" segments of input. A correction is applied to the data buffer every other baud to align the program timing to the timing of the received input. If baud transition indications appear in both the "early" and "late" intervals (as might happen when the demodulator first receives a signal), a unidirectional synchronization correction is applied until the baud transition is found in only one of the intervals.

### 3.3.2 3600/4800 bps TDPSK, No Coding

Table 3.3-1 shows the parameters of the 3600/4800 bps TDPSK program.

Figure 3.3-3 shows the block diagram of the 3600-b/s and 4800-b/s modes of the uncoded TPSK program. In this diagram, the preamble generation and detection routines are shown connected to the transmit D/A and receive A/D by a switch to indicate that they only operate during the setup of a link. The operation of these algorithms is discussed in subsection 4.1.

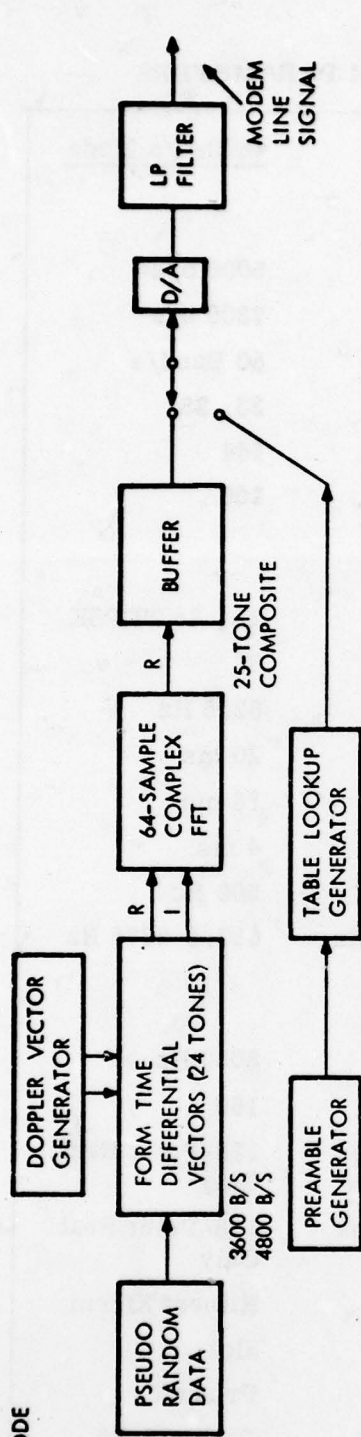
In the transmitter, a 64-sample complex FFT is used to generate the composite tone set. In this scheme, the real and imaginary part of each tone is generated by adding the appropriate phase shift to the previous transmitted vectors. The composite frequency domain signal is fed to the FFT which in turn produces a composite time domain line signal.

The primary difference between the 3600 and 4800-b/s mode receiver and the 2400-b/s mode receiver is that the higher data rate modes use a Hilbert Transform generator to correct for Doppler offset. With this scheme, the orthogonal outputs of the Hilbert Transform generator are multiplied by a cos and sin function whose frequency is equal to the measured offset. The sum of multiplier outputs is a real signal which has been shifted by the appropriate correction frequency. Since the input is real, only the real part of the demodulation FFT is loaded.

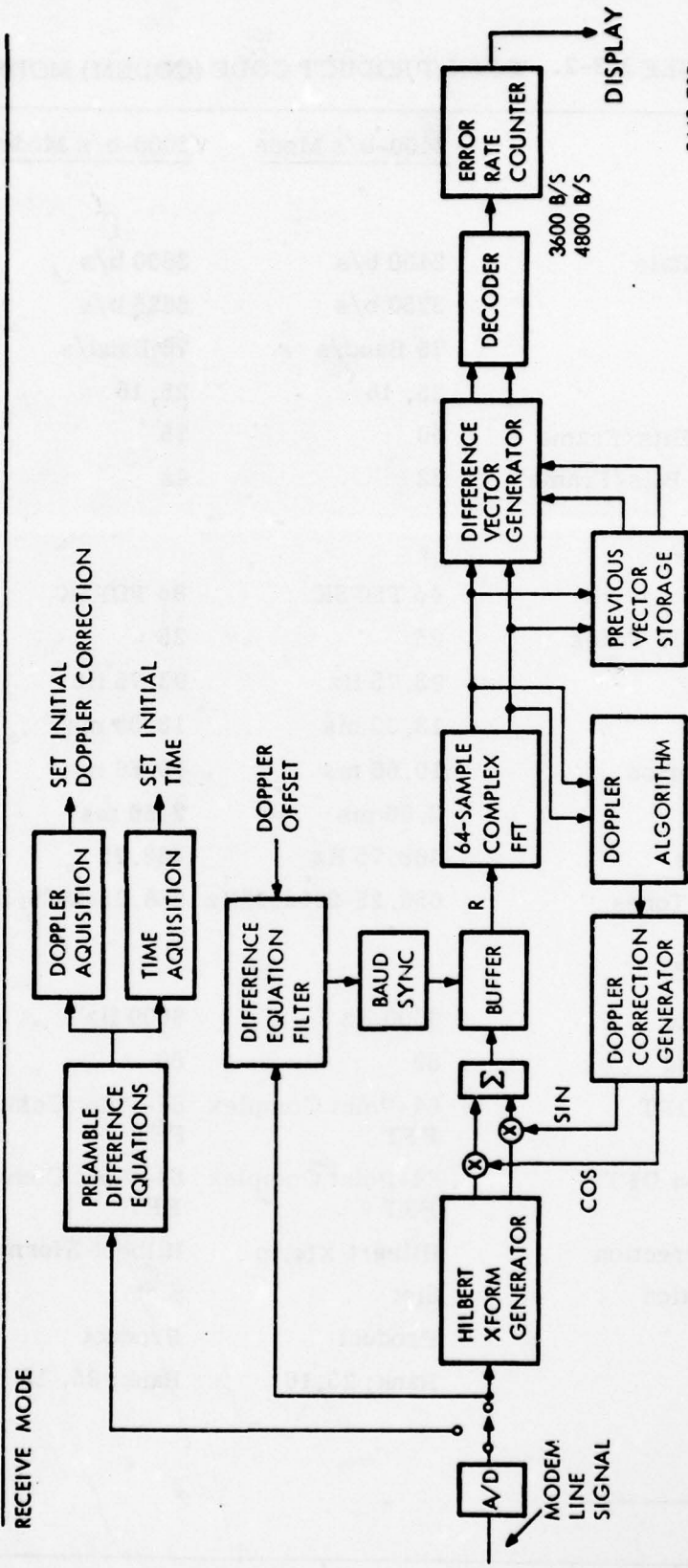
**TABLE 3.3-2. TDPSK/PRODUCT CODE (CODEM) MODEM PARAMETERS**

	<u>2400-b/s Mode</u>	<u>3600-b/s Mode</u>	<u>4800-b/s Mode</u>
<b><u>DATA</u></b>			
Information Rate	2400 b/s	3600 b/s	5000 b/s*
TX. Rate	3750 b/s	5625 b/s	7200 b/s
Frame Rate	75 Baud/s	75 Baud/s	50 Baud/s
Coding	25, 16	25, 16	36, 25
Number TX Bits/Frame	50	75	144
Number Inf. Bits/Frame	32	48	100
<b><u>SIGNALLING</u></b>			
Modulation	4 $\phi$ TDPSK	8 $\phi$ TDPSK	2A, 8 $\phi$ TDPSK
Number of Inf. Tones	25	25	36
Tone Spacing	93.75 Hz	93.75 Hz	62.5 Hz
Baud Period	13.33 ms	13.33 ms	20 ms
Orthogonal Period	10.66 ms	10.66 ms	16 ms
Guard Time	2.66 ms	2.66 ms	4 ms
Doppler Tone	468.75 Hz	468.75	500 Hz
Information Tones	656.25-2906.25Hz	656.25-2906.25Hz	687.5-2875 Hz
<b><u>PROCESSING</u></b>			
Sampling Rate	6000 Hz	6000 Hz	8000 Hz
Samples/Baud	80	80	160
Modulation DFT	64-Point Complex FFT	64-Point Complex FFT	128-Point Real Only
Demodulation DFT	64-Point Complex FFT	64-Point Complex FFT	128-Point Real Only
Doppler Correction	Hilbert Xform	Hilbert Xform	Hilbert Xform
Synchronization	Slot	Slot	Slot
Encoder	Product	Product	Product
Decoder	Rank; 25, 16	Rank; 25, 16	Rank; 36, 25
* See Text			

TRANSMIT MODE



RECEIVE MODE



5445-73E

Figure 3.3-3: TDPSK 1/2 Duplex Modem, 3600 b/s and 4800 b/s Modes

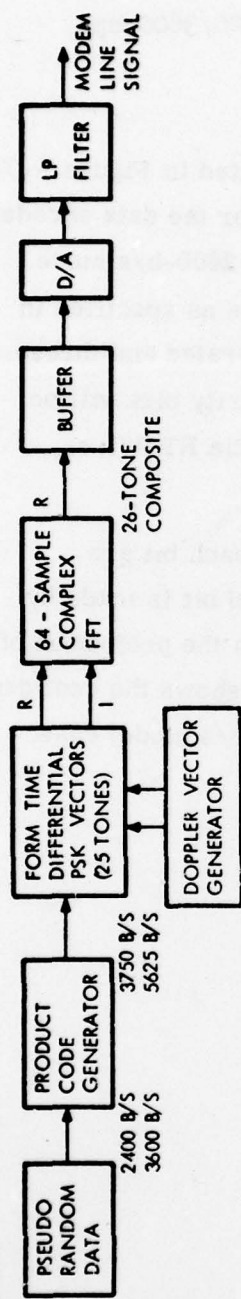
### 3.3.3 2400/3600 bps TDPSK With Coding

Table 3.3-2 describes the parameters of the 2400/3600 bps CODEM Modem.

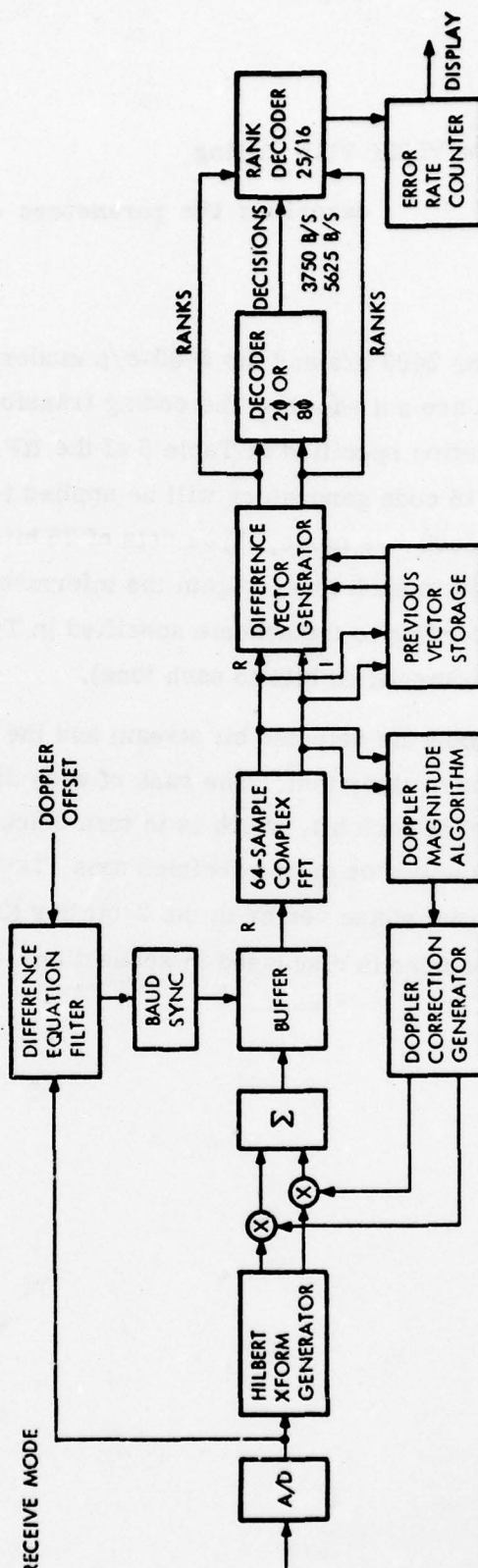
A block diagram of the 2400 b/s and the 3600-b/s modes is presented in Figure 4.3-4. Again the preamble routines are not shown. The coding transformation for the data encoder will be the coding transformation specified in Table 5 of the RFP. In the 2400-b/s mode one bit from each of the 25, 16 code generators will be applied to each tone as specified in Table 6 of the RFP. In the 3600-b/s mode, three sets of 25 bits are generated and three bits of modulation are applied to each tone. Again the information and parity bits will be applied to the data tones according to the scheme specified in Table 6 of the RFP (i.e., one bit from each set of 25 transmitted bits to each tone).

In the demodulator, both the detected bit stream and the rank for each bit are transmitted to the rank decoder algorithm. The rank of each demodulated bit is made by ordering the confidence level of each bit, which is in turn calculated from the projection of the received differential phase vector on the decision axis. Figure 3.3-5 shows the confidence levels for arbitrary differential phase vector in the 2-bit per tone (2400-b/s mode) case. The operation of the rank decoder is discussed in subsection 4.3.

## TRANSMIT MODE

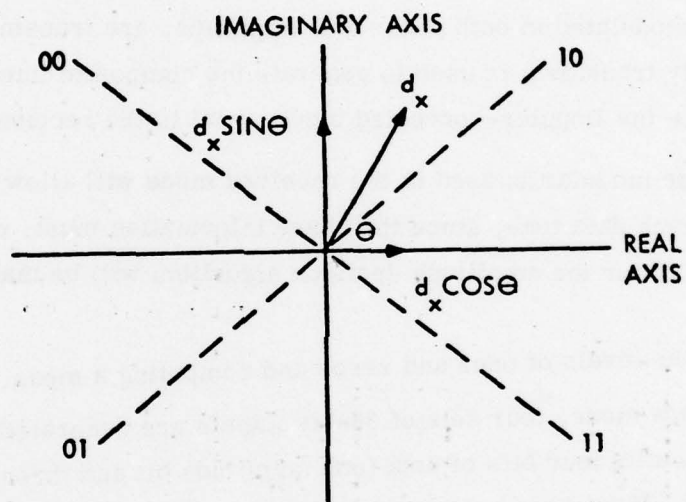


## RECEIVE MODE



5450-73E

Figure 3.3-4: TDPSK - Product Code (Codem), 1/2 Duplex Modem, 2400 b/s and 3600 b/s Modes



$d_x$  DECODED AS 10  
 CONFIDENCE LEVEL OF 1 =  $d_x \cos \Theta$   
 CONFIDENCE LEVEL OF 0 =  $d_x \sin \Theta$

5451-73E

**Figure 3.3-5** Confidence Levels for Arbitrary Differential Phase  
 Vector in 2-Bit Per Tone (2400 b/s Mode) Case

#### 3.3.4 4800 bps Codem and DEFT

Table 3.3-2 describes the parameters of the 4800 bps Codem Modem.

As shown in Table 3.3-2 the actual information rate of the 4800-b/s mode is 5000 b/s (100 decoded bits/frame times 50 frames/second). An exact data rate of 4800 b/s could be achieved by simply lowering the baud rate to 48 bauds/second. Such a change would increase the guard time to 4.83 milliseconds.

Figure 3.3-6 shows the block diagram of the 4800-b/s mode. In this mode 36 information tones, modulated in both phase and amplitude, are transmitted each frame. A 128-sample real only transform is used to generate the composite line signal in the transmitter and to analyze the Doppler-corrected input signal in the receive mode.

The amplitude modulation used in the received mode will allow one of two levels to be transmitted for each data tone, since the phase information must, of course, be maintained. The threshold for the amplitude decision algorithm will be made adaptive by averaging the previous levels of ones and zeros and computing a mean.<sup>10</sup>

In the 4800-b/s mode, four sets of 36-bit outputs are generated by the data encoder. Since each tone transmits four bits of data (one amplitude bit and three phase bits), one bit of data from each 25, 36 encoder set will be applied to each of the 36 data tones. The ranks for the DPSK bits will be computed in the same manner as the ranks for the 3600-b/s mode. The amplitude bit ranks will be computed by ordering the distance between the received amplitude and the decision threshold for each bit.

AD-A074 321

GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 9/2  
HF PROGRAMMABLE MODEM.(U)  
OCT 74

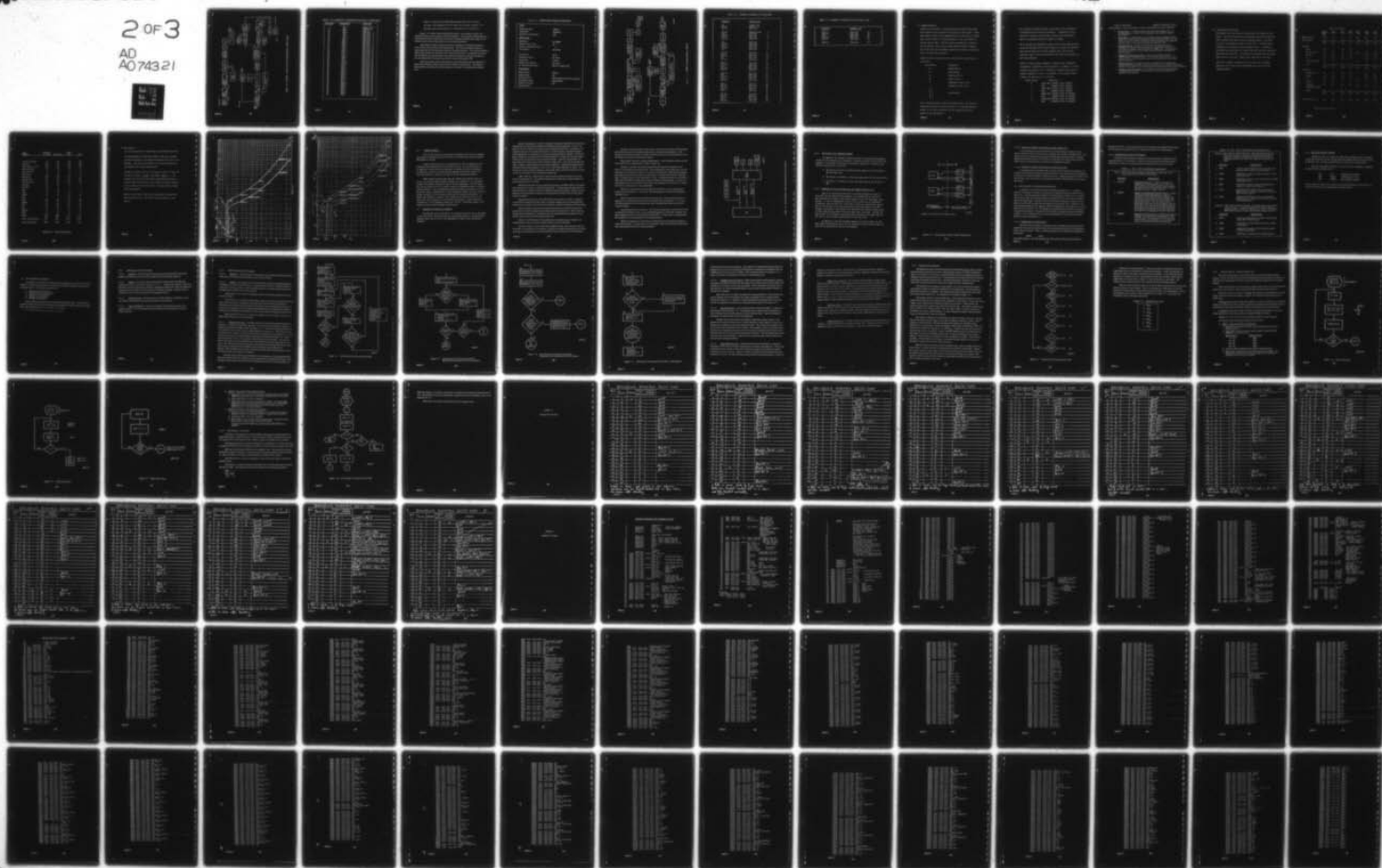
N00014-74-C-0299

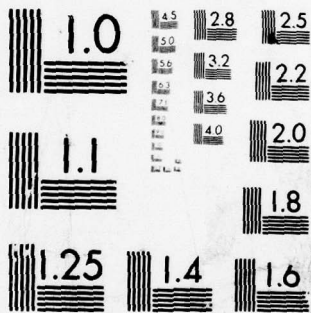
NL

UNCLASSIFIED

2 OF 3

AD  
A074321





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

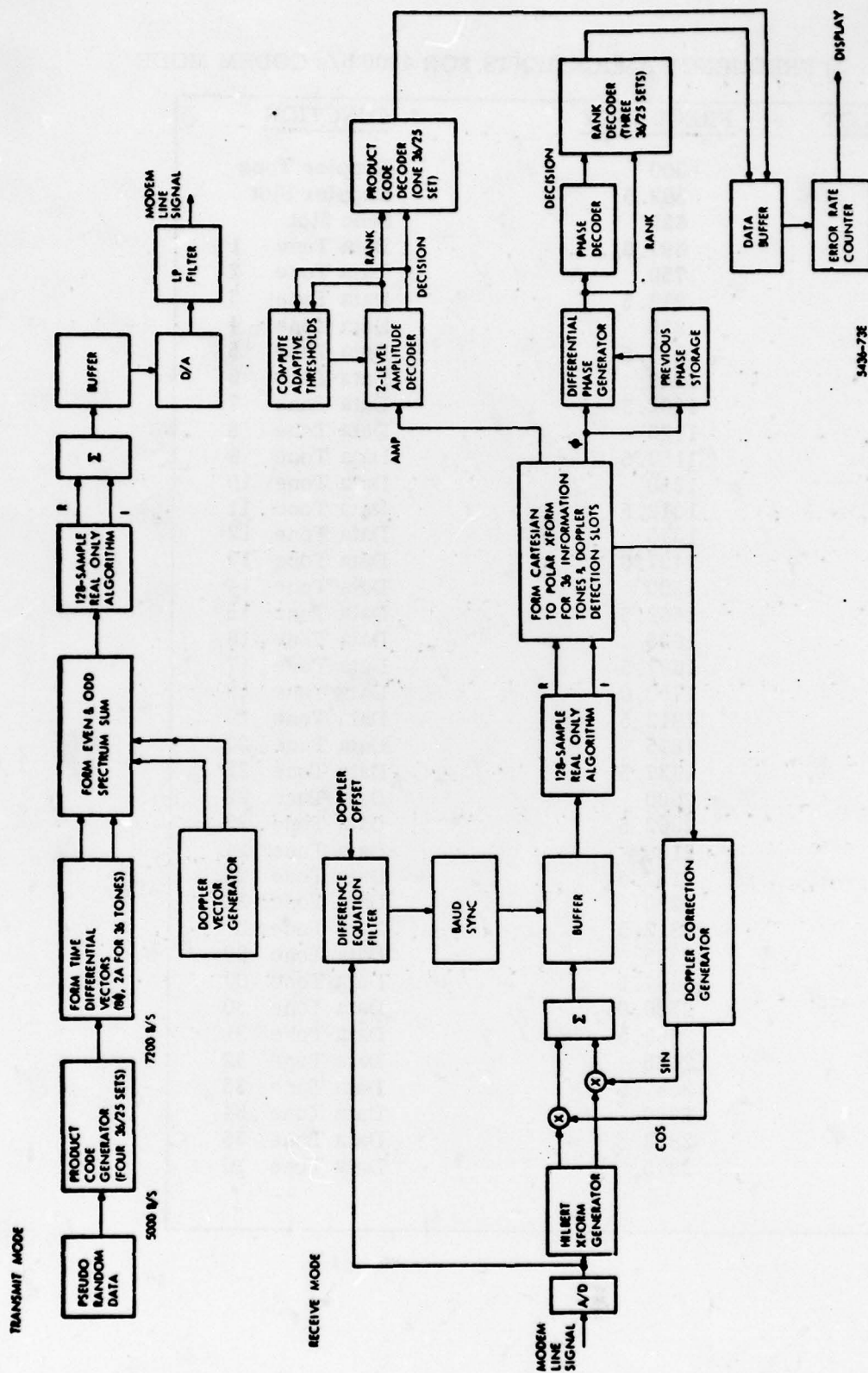


Figure 3.3-6: TDPSK - Product Code 1/2 Duplex Modem, 4800 b/s Mode

TABLE 3.3-3: FREQUENCY ASSIGNMENTS FOR 4800 b/s CODEM MODE

<u>FFT SLOT</u>	<u>FREQUENCY</u>	<u>FUNCTION</u>
8	500	Doppler Tone
9	562.5	Doppler Slot
10	625	Sync Slot
11	687.5	Data Tone 1
12	750	Data Tone 2
13	812.5	Data Tone 3
14	875	Data Tone 4
15	937.5	Data Tone 5
16	1000	Data Tone 6
17	1062.5	Data Tone 7
18	1125	Data Tone 8
19	1187.5	Data Tone 9
20	1250	Data Tone 10
21	1312.5	Data Tone 11
22	1375	Data Tone 12
23	1437.5	Data Tone 13
24	1500	Data Tone 14
25	1562.5	Data Tone 15
26	1625	Data Tone 16
27	1687.5	Data Tone 17
28	1750.0	Data Tone 18
29	1812.5	Data Tone 19
30	1875	Data Tone 20
31	1937.5	Data Tone 21
32	2000	Data Tone 22
33	2062.5	Data Tone 23
34	2125	Data Tone 24
35	2187.5	Data Tone 25
36	2250	Data Tone 26
37	2312.5	Data Tone 27
38	2375	Data Tone 28
39	2437.5	Data Tone 29
40	2500.0	Data Tone 30
41	2562.5	Data Tone 31
42	2625	Data Tone 32
43	2687.5	Data Tone 33
44	2750.	Data Tone 34
45	2812.5	Data Tone 35
46	2875.	Data Tone 36

Table 3.3-4 describes the FDPSK modem program which will only have one mode. The frequencies for the tone set are shown in Table 3.3-5. The data tones are referenced to a reference tone at 515.625 Hz.

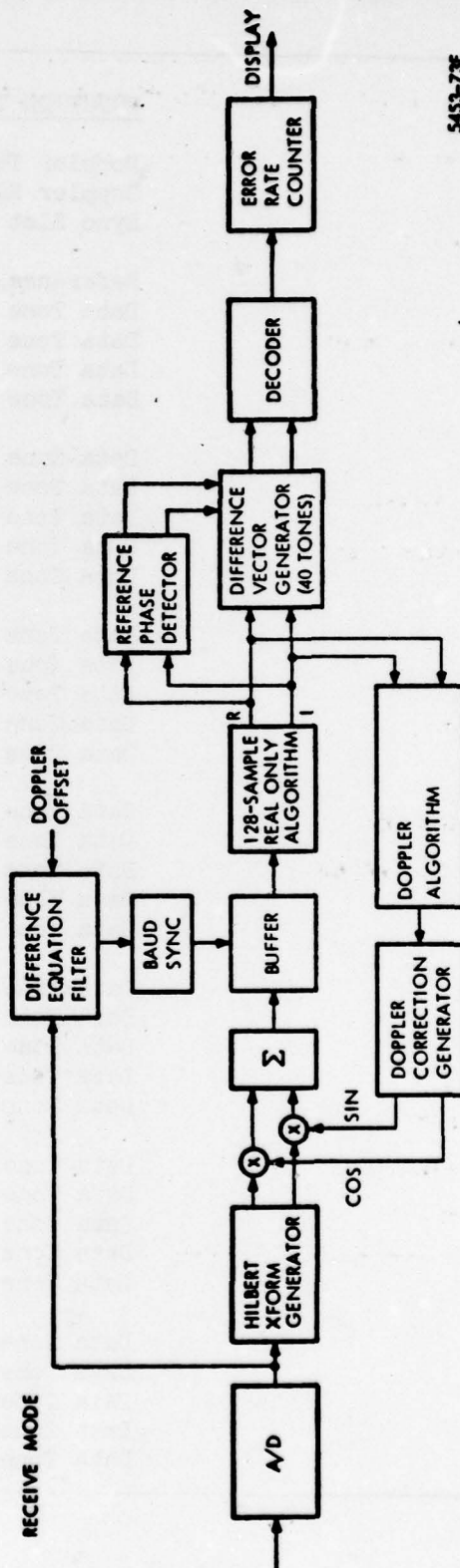
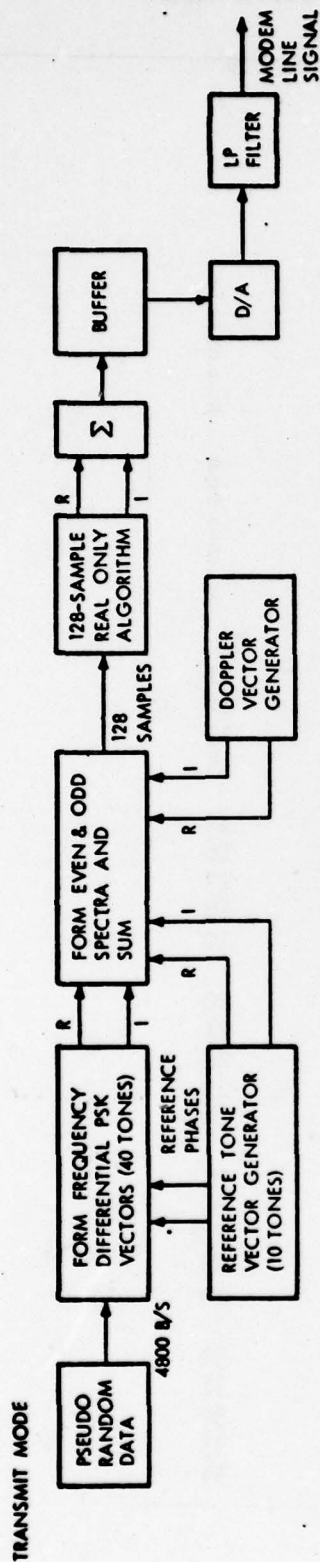
Figure 3.3-6 shows a block diagram of this mode. For the sake of clarity, the preamble functions are not shown in this block diagram. In the frequency differential phase shift keyed system the complex data tone vectors are phase modulated with respect to ten reference tones which are generated from the reference generator.

The complex frequency domain vectors are expanded in the frequency domain to form an even real part and an odd imaginary part. The real and imaginary parts are summed and fed into the 128-sample real only transform algorithm. The real and imaginary parts of the output of this transform are in turn summed to form the composite line signal. The summing at the input and output of the transform utilizes the symmetric properties of a real valued DFT to eliminate the need of a complex 128-sample FFT.

The receive mode of this modem uses the same 128-sample real only algorithm to generate the frequency domain vectors for the reference and data tones. The signal is demodulated by computing the data tone phases to the phase of the appropriate reference tone.

Table 3.3-4: **FDPSK (DEFT) MODEM PARAMETERS**

<b><u>DATA</u></b>	
Information Rate	4800 b/s
Frame Rate	40 Baud/s
Number Inf. Bits/Frame	120
<b><u>SIGNALLING</u></b>	
Modulation	8 $\phi$ FDPSK
Number of Inf. Tones	40 -
Number of Reference Tones	1
Tone Spacing	46.875 Hz
Baud Period	25 ms
Orthogonal Period	21.33 ms
Guard Time	3.66 ms
Doppler Tone Spacing	468.75 Hz
Inf. and Reference Tones	609.375 - 2906.25 Hz
<b><u>PROCESSING</u></b>	
Sampling Rate	6000 Hz
Samples/Baud	150
Modulation DFT	128-Point Real Only FFT Algorithm
Demodulation DFT	Hilbert Xform
Doppler Correction	Slot
Synchronization	



5453-73E

Figure 3.3-6: FDPK 1/2 Duplex Modem, 4800 b/s

Table 3.3-5: FREQUENCY ASSIGNMENTS FOR FDPSK MODEM

<u>FREQUENCY</u>	<u>FUNCTION TONE</u>	
375	Doppler Tone	
421.875	Doppler Slot	
468.75	Sync Slot	
515.625	Reference Tone	1
562.5	Data Tone	1
609.375	Data Tone	2
656.25	Data Tone	3
703.125	Data Tone	4
750	Data Tone	5
796.875	Data Tone	6
843.75	Data Tone	7
890.625	Data Tone	8
937.5	Data Tone	9
984.375	Data Tone	10
1031.25	Data Tone	11
1078.125	Data Tone	12
1125.	Data Tone	13
1171.875	Data Tone	14
1218.75	Data Tone	15
1265.625	Data Tone	16
1312.5	Data Tone	17
1359.375	Data Tone	18
1406.25	Data Tone	19
1453.125	Data Tone	20
1500	Data Tone	21
1546.875	Data Tone	22
1593.75	Data Tone	23
1640.625	Data Tone	24
1687.5	Data Tone	25
1734.375	Data Tone	26
1782.25	Data Tone	27
1828.125	Data Tone	28
1875.0	Data Tone	29
1921.875	Data Tone	30
1968.75	Data Tone	31
2015.625	Data Tone	32
2062.5	Data Tone	33
2109.375	Data Tone	34

**TABLE 3.3-5: FREQUENCY ASSIGNMENTS FOR FDPSK MODEM (Cont.)**

<u>FREQUENCY</u>	<u>FUNCTION</u>	<u>STONE</u>
2156.25	Data Tone	35
2203.125	Data Tone	36
2250	Data Tone	37
2296.875	Data Tone	38
2343.75	Data Tone	39
2390.625	Data Tone	40

### 3.4 Software Operation

Control of the programs is originated from the modem control panel on the manual switch register on the processor control panel. After the program is read into the machine either from the PDP-11 or paper tape and the initial or starting location is set, the machine is put into the run mode. From this point on operation is controlled by the switches. Each of the programs varies slightly on control due to the functional difference between modes; however, standardization is provided where possible.

Display controls originate from the manual switch register and are given below:

Bit Location	Description
15	Random Data In
14	Error Control
11	Anything, Low 11
10	Input, Diff Eq
9	Unprocessed Vector, Low 6
8	Difference Vector, Low 6
5 4 3 2 1 0	Tone Location

Bit 15 switches between internal and external data. Bit 14 in the Codem modes switches in the error decoder. In the mode where the decoder is not used, the parity bits are transmitted but just ignored in the demodulation.

Bit 11 allows any memory location in the lower 2K of data memory to be displayed on one of the D/A converters. A capability is thus provided to display flags and other locations where unprocessed data can be inspected while the program is running.

Bit 10 displays the difference equation used for time synchronization on one D/A converter and the input samples on a second D/A converter. Bits 9 and 8 output the real and imaginary axis of the FFT and processed output of the tone selected by the lower 6 bit locations on the switch register.

Control of which program, internal or external clock, preamble or no preamble is exercised by the mode control. In general - bit one controls the program, bit 2 is internal or external clock and bit 3 controls preamble on or off. For instance, in the Codem III/DEFT program, the mode switch is as follows:

Switch Setting	Description
0	CODM 4800, INTERNAL CLOCK, PREAMBLE
1	DEFT , INTERNAL CLOCK, PREAMBLE
2	CODM 4800, EXTERNAL CLOCK, PREAMBLE
3	DEFT , EXTERNAL CLOCK, PREAMBLE
4	CODM 4800, INTERNAL CLOCK, NO PREAMBLE
5	DEFT , INTERNAL CLOCK, NO PREAMBLE
6	CODM 4800, EXTERNAL CLOCK, NO PREAMBLE
7	DEFT , EXTERNAL CLOCK, NO PREAMBLE

In all of the programs a number of parameters can be easily modified. The most significant of these variable parameters are listed below:

- a. Sampling Rate - A single parameter will control the sampling rate. As the sampling rate is varied, the tone spacings, tone positions, and data rate will vary.
- b. Frame Rate - A single parameter, the number of input samples per frame, will vary the frame rate and consequently the guard time and data rate. The lower limit to this variation will be the number of samples in the orthogonal set (64 or 128). In this case, the guard time would go to zero.
- c. The Tone Position - The position of each tone in the data set will be referenced to a stored table in data memory. Modification of this table will result in frequency shifts in the data or Doppler tones. Since the tones must fall on an FFT slot, such a shift can only be made in increments of the tone spacing.
- d. Doppler and Time Tracking Rates - The rates of frequency and time tracking will be controlled by four or five locations in data memory. The modification that these parameters make to the tracking rates will be identified in the listings.
- e. Number of Tones - The number of tones in the data set in the modes which do not require a fixed number of data bits per frame for the coding scheme will be variable. This number will be varied by adjusting several parameters in data memory.
- f. Number of Preamble Frames - The number of preamble frames transmitted and expected in the receiver will be varied from 5 to 32 by modifying two locations of data memory.

### 3.5 Program Memory and Timing

The programs have been written to operate both half duplex and full duplex. Two of the seven modes at the present do not have sufficient time to operate full duplex with all functions enabled. The times for all of the programs are given in Table 3.5-1. The times are a combination of calculated times and measured times. It should be noted that the times given are nominal times and worst case conditions can expand the noted time. For instance, the decoder when correcting bursts of error can take a longer period than noted in the chart.

Table 3.5-2 presents the number of 32 bit words used in program memory and 16 bit words used in data memory for each of the four combined programs.

TIMES IN MILLISECONDS

	Program CODEM 4800	Timing DEFT	CODEM 3600	CODEM 2400	TDPSK 2400	TDPSK 3600	TDPSK 4800
Sample Interrupt	.96	.96	.5	.5	5.2	0.5	0.5
Data Interrupt	1.5	1.5	.75	.5	.56	.84	1.26
	2.46	2.46	1.25	1.0	5.76	1.34	1.76
Modulation							
Encode	.05	-	.05	.05	-	-	-
Form Mod. Vectors	1.1	.1.1	.8	.8	0.5	0.8	0.8
Scramble	.34	.34	.34	.34	-	0.34	0.34
FFT	1.6	1.6	1.6	1.6	-	1.6	1.6
Even/Odd Separation	.6	.6	-	-	-	-	-
Store	.4	.4	.2	.2	-	0.2	0.2
	4.09	4.04	2.99	2.99	0.5	2.94	2.94
Demodulation							
Hilbert Transform	2.2	2.2	1.4	1.4	-	1.4	1.4
Scramble	.34	.34	.34	.34	-	0.34	0.34
FFT	1.6	1.6	1.6	1.6	1.6	1.6	1.6
Even/Odd	.6	.6	-	-	-	-	-
Demodulate (Includes Bank, Sig Proc Doppler Det)	5.0	2.8	3.0	2.0	0.8	1.2	1.4
Decode	4.5	-	2.6	1.8	-	-	-
	14.24	7.54	8.94	7.14	2.4	4.54	4.74
Total Full Duplex Time	20.79	14.04	13.18	11.13	8.66	8.82	9.02

TABLE 3.5-1: Timing Estimates

CODE ROUTINE	Codem III + DEFT	Codem I/II	3600/ 4800	2400
Output Mod Sample	43	37	40	101
Preamble	379	372	383	389
Input Sample + Sync	129	130	138	187
Data Interrupt	85	83	92	61
3rd Interrupt	4	4	4	4
Wait Loop	34	36	22	30
Modulation	147	89	92	--
Demodulation	411	437	303	290
Initialization	115	122	111	47
Scramble	30	30	30	--
Even/Odd	53	--	--	--
Erec.	28	28	--	--
PRCG	13	13	13	--
DFRAM	10	10	18	58
NOPS	--	--	20	--
CTOP	159	159	159	159
INVERT	30	30	30	30
FFT	185	185	185	185
VMP	18	18	18	18
SIN/COS	13	13	13	13
DISP	50	50	50	55
DECOD	<u>411</u>	<u>639</u>	<u>--</u>	<u>--</u>
Total Program Memory	2347	2485	1734	1632
Total Data Memory	2899	2662	1768	1124

TABLE 3.5-2: Memory Requirements

### 3.6 Test Results

All of the programs were operationally and functionally tested.

The rank decoder was extensively tested to the error patterns inclosed in Appendix A and operated successfully with all of the patterns. This test was performed on the decoder operating as an integral block of the modem program.

Frequency responses were run on the input and output filters and are given in Figure 3.6-1 and Figure 3.6-2. Two responses are given since the design incorporated provisions to rapidly change the bandwidth of the filters by modular change of a set of eight resistors per filter. Two modules were provided with the equipment.

Signal to noise tests, fading tests and doppler tracking tests were performed at NRL. The results are not included in this report.

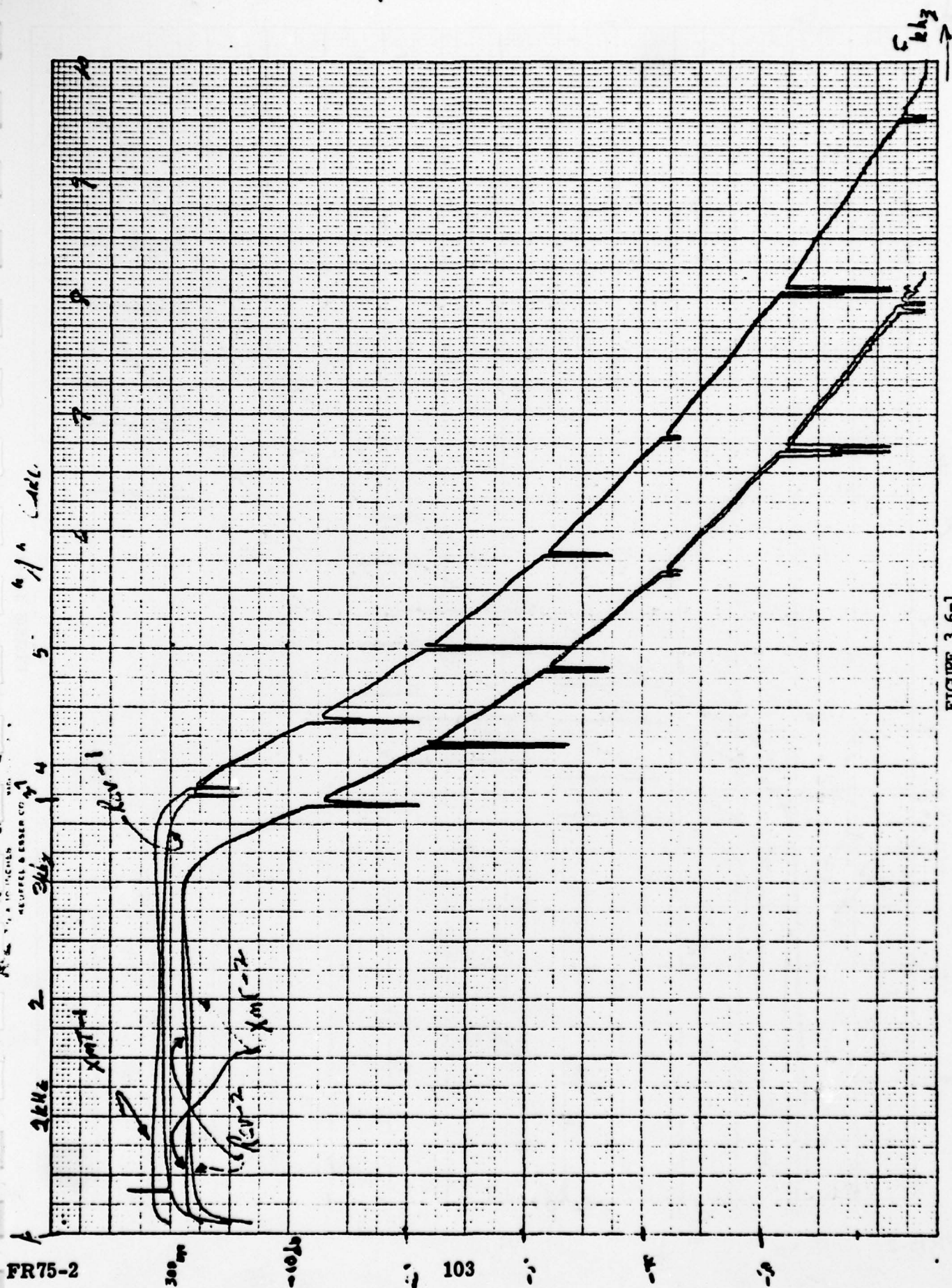


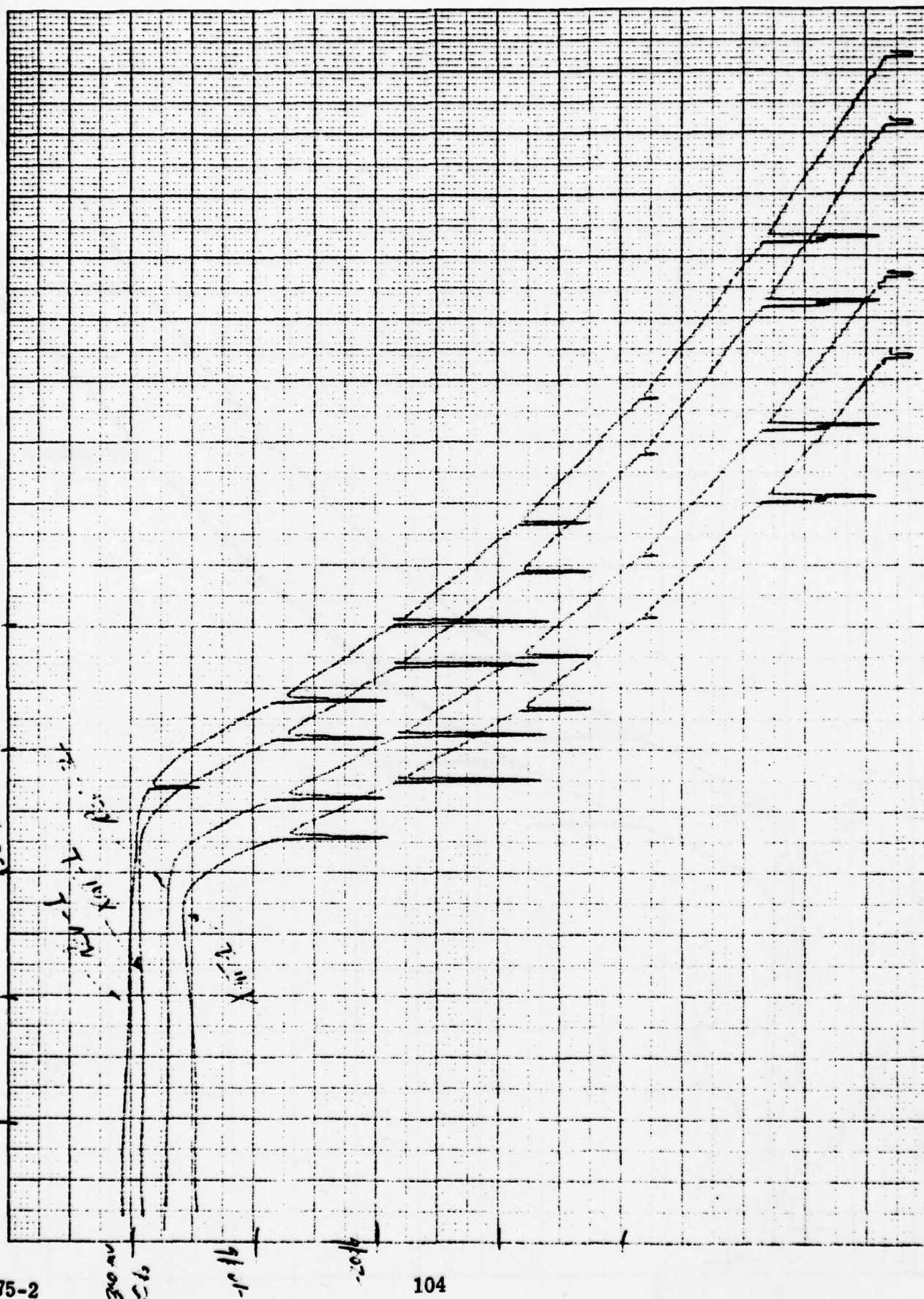
FIGURE 3.6-1

Feb 7

10 X 10 TO 1 1/2 INCH 46 1472

COGNITION 71(10) 1003-1013

LB-241



**FIGURE 3.6-2**

#### 4.0 SUPPORT PROGRAMS

This section describes the software development both for the assembly and loading of programs from the PDP 11/45 to the PSP and diagnostic programs developed for the PSP.

Signal processing tasks are performed on the PSP by appropriate PSP programs written specifically for the task at hand. In this environment it is important to have efficient techniques to develop and test PSP programs. An extensive collection of field-proven hardware and software support systems was provided to simplify the task of the PSP program development.

These support systems range from the simple PSP assembler which can run on any suitable general-purpose computer to a complete set of peripheral hardware and support software for use by the PSP programmer. The PSP assembler can be oriented to run in either a batch or time-shared mode on a general-purpose computer. For the laboratory requiring peripheral equipment, one or more PSP's may be integrated with a PDP-11 computer system, such that the PSP's become the masters of the system controlling the available peripherals. To support the PSP/PDP-11 system at NRL, assembler was provided to run on the PDP-11 with 20K of memory plus an extensive collection of support software including an on-line debugging program for the PSP, a PSP dump program, a PSP loader and other useful packages itemized below.

##### 4.1 PHILOSOPHY OF PSP/PDP-11 APPROACH

The efficient use of the PDP-11 in conjunction with the PSP is discussed in this subsection, which describes the relationship between a general-purpose computer system and the PSP in the development of modern signal processing equipment.

By way of background, the PSP is designed to provide an improvement over previous methods both in developing and implementing signal processing equipment. Prior to the advent of the PSP, sophisticated signal processing equipment was built in two phases, the simulation phase and the implementation phase. In the first phase, proposed techniques were programmed "or simulated" on a large scale general-purpose computer. During this phase, parameters of design could be varied and analyzed. Through this simulation effort a final system was specified and all parameters defined. During the second phase, the final system was constructed through a hardware design effort. Once the second phase had begun, changes in design and parameters became expensive. Sophisticated equipment was excessively difficult and expensive to build and in some cases impossible to debug. The success of a system could be impaired in cases where subtle variations between the simulation and the end product caused unexpected degradations.

Using a PSP as a replacement for special-purpose hardwired designs overcomes most of these difficulties. The hardware design effort of phase two is largely replaced by a programming effort on the PSP.

Making the transition from the simulation program to the PSP program is far easier than making the transition to a hardware design effort. Communication and documentation problems between a software effort and a hardware effort are eliminated. Indeed, the programmers who developed the simulation software generally follow on with the PSP implementation program.

The programming effort on the PSP becomes the last effort of design. Flexibility is maintained throughout the design effort, and system changes may be incorporated with minimum cost even after systems have been delivered, simply by changing the PSP program.

With the flexibility during the PSP programming effort the distinction between the simulation phase and the implementation phase quickly becomes blurred. Many of the studies previously performed in a simulation effort on a general-purpose computer are more quickly achieved during the PSP programming effort.

Indeed, if the PSP had all of the capabilities of a general-purpose computer, the simulation effort using a separate computer would be entirely eliminated.

The PSP, however, does not have these capabilities and, since a primary objective of the PSP is to provide low cost hardware to perform the signal processing task in the field, its design is not burdened by facilities and software that are useful only in the laboratory.

In order to achieve the best of both worlds - the minimum cost deliverable hardware, and the facilities that are desirable during system simulation and development - GTE Sylvania has interfaced the PSP to a PDP-11 in such a way that the PSP programmer has at his disposal the facilities of the PDP-11 system.

The hardware interface is quite straightforward. Three data paths between the PSP and the PDP-11 are implemented as shown in Figure 4-1.

The path to the PSP card reader input allows PSP programs assembled on the PDP-11 to be loaded into the PSP directly. This eliminates the card reader and avoids the unnecessary step of punching cards. As a result of this data path, PSP assembly language programs may be read into the PDP-11 card reader, or written at the PDP-11 typewriter input. Programs may be stored on the PDP-11 disk. The powerful facilities of the PDP-11 editor and file handling software (PIP) will be available to the programmer. The fact that the programs are run on the PSP yet are written and assembled on the PDP-11 is a distinction that is of little concern to the programmer.

With the I/O data paths between the PSP and the PDP-11, data is easily passed between the two processors. As a result, the PSP program has the PDP-11 peripherals available for use.

Data to be processed may originate from a tape unit, results may be printed on a typewriter and program parameters may be requested from the typewriter. As a result, the PSP programmer has the hardware and control normally associated with a general-purpose computer.

The programmer has two choices. He could write programs in Fortran which would be executed by the PDP-11 CPU, or he could write programs in PAP (PSP Assembly Program language) which would be executed by the PSP CPU. In either case, the programmer has available all of the peripherals of the combined facility.

With this arrangement, the need for the traditional Fortran simulation program is reduced considerably. In many cases, the "simulation" effort can be done more effectively with a PSP program than with a Fortran program.

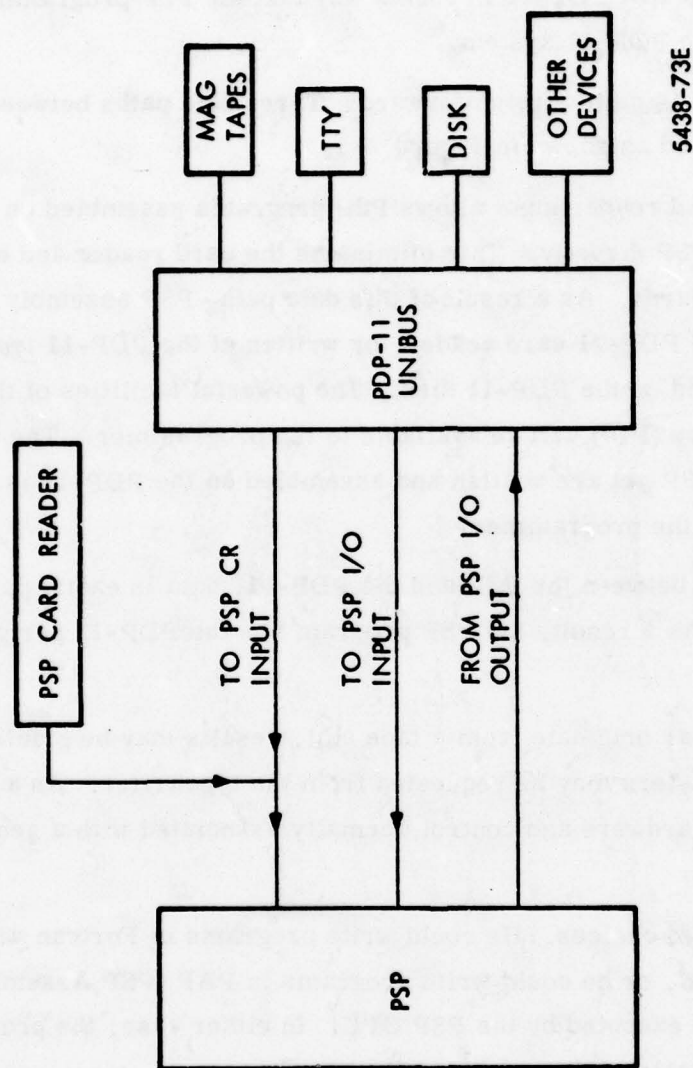


Figure 4-1. Block Diagram of Data Paths Between PSP and PDP-11

#### 4.1.2 PSP TO PDP 11/45 INTERFACE HARDWARE

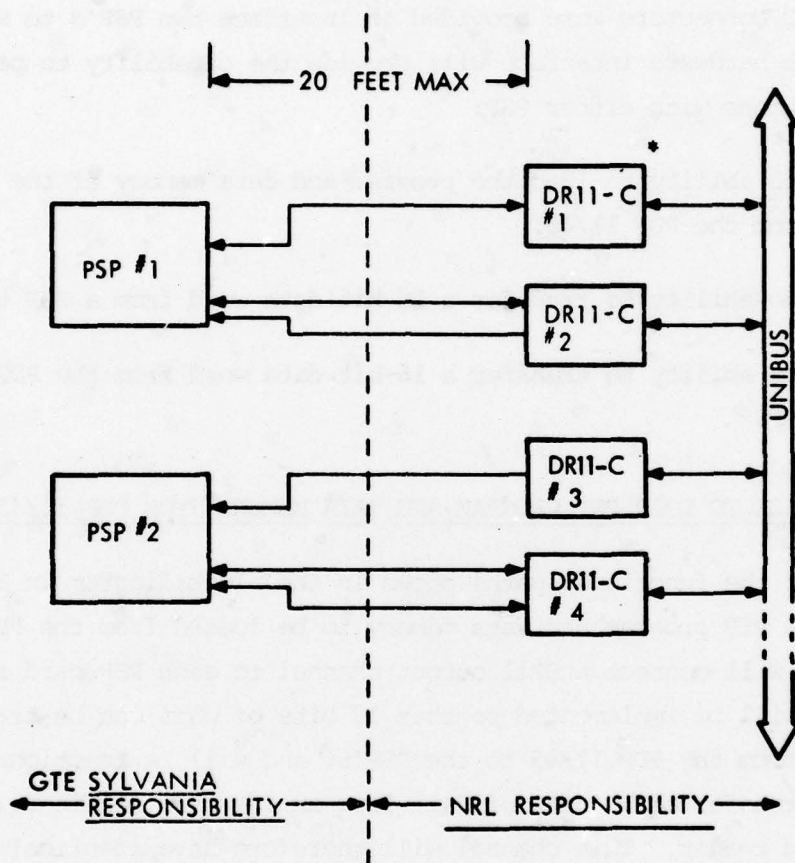
To implement the PSP/PDP-11 System outlined in the preceding paragraphs, cables and I/O connectors were provided to interface two PSP's to NRL's PDP 11/45 computer. The hardware interface will provide the capability to perform three distinct functions with either PSP:

- a. The ability to load the program and data memory of the PSP directly from the PDP 11/45.
- b. The ability to transfer a 16-bit data word from a PSP to the PDP 11/45.
- c. The ability to transfer a 16-bit data word from the PDP 11/45 to a PSP.

##### 4.1.2.1 INTERFACE TO LOAD PSP PROGRAM AND DATA MEMORY FROM PDP 11/45

One of the functional paths shown in the block diagram in Figure 4-2 will allow the PSP program and data memory to be loaded from the PDP 11/45. The interface will connect a DR11 output channel to each PSP card reader input channel. It will be implemented so that 12 bits of data can be transferred, in parallel, from the PDP 11/45 to the PSP's, and will be functionally equivalent to the transfer of 12 bits of data from an IBM card column to the PSP through a card reader. This channel will therefore have identical capabilities to those otherwise provided by the card reader input to the PSP. The PSP will then be able to be loaded by transferring PSP binary deck card images from the PDP 11/45 to the PSP under program control of the PDP 11/45.

In addition to the data transfer capability of the interface, the PSP card reader enable/disable signal will be fed back to the PDP 11/45 so that its state can be determined by the PDP 11/45 program.



341-73E

\* THESE MAY OPTIONALLY BE DR11-A UNITS.

Figure 4-2. Block Diagram of PSP-to-PDP 11/45 Interface

#### 4.1.2.2 Interface to Transfer 16-Bit Data from PSP to PDP 11/45

A second data path to each PSP shown in the block diagram in Figure 2-10 will allow the PSP's to transfer 16 bits of data in parallel from the PSP to the PDP 11/45. This transfer will be accomplished by execution of a PSP output instruction (OUTA or OUTM). Signals generated by execution of the OUTA or OUTM instruction will be used to alert the PDP 11/45 that it must perform an instruction to complete the transfer.

#### 4.1.2.3 Interface to Transfer 16-Bit Data from PDP 11/45 to PSP

A third data path to each PSP shown in Figure 2-10 will also allow a PSP program to obtain 16 bits of data in parallel from the PDP 11/45. This transfer will be accomplished by execution of a PSP input (INPA or INPM) instruction. Signals generated by the PDP 11/45 will be used to alert the PSP that it must perform an INPA or INPM instruction to perform the transfer.

### 4.1.3 PSP-TO-PDP 11/45 INTERFACE SOFTWARE

In addition to the interface hardware described in subsection 4.2, interface software is necessary to realize the potential of the interface between the PSP's and the PDP 11/45. With the appropriate software in both the PSP and the PDP 11/45, the PSP's will be able to make use of the peripherals on the PDP 11/45 for either input or output as desired and appropriate. Similarly, PDP 11/45 software is required to make use of the ability to load the PSP program and data memory from the PDP 11/45. Generally, each I/O exchange between the two computers requires a matching pair of programs - one in the PSP and one in the PDP 11/45. The software to implement these functions is referred to as the interface software. Software will be delivered to implement the functions detailed in Table 4-1. It is divided into three groups corresponding to the three functional data paths described in subsection 4.2.

#### 4.1.3.1 Implementation of PSP Software

The PSP interface software is included as part of the PDP-11 version of the PSP assembler so that the programmer need not be concerned with the details of the programming to use the interface. That is, the PSP programmer will include such statements as, for example,

RWND      TAPE

in his PSP program. The PSP assembler will then insert the PSP coding necessary to

perform the function. Each such instruction will be treated by the assembler as a request for the appropriate predefined "Macro" routines to be inserted in the PSP program.

#### 4.1.3.2 Implementation of PDP 11/45 Software

The PDP 11/45 I/O interface software in Groups II and III are combined into one complete program to be executed in conjunction with the execution of the PSP Program. The software in Group I are separate programs reflecting their distinctive functions.

TABLE 4-1. PSP - PDP 11/45 INTERFACE SOFTWARE

Group I - PDP 11/45 to PSP Program and Data Memory Load Group. This group operates with the PSP in the halt mode.

<u>PROGRAM</u>	<u>DESCRIPTION</u>
1. PSPLDR	<u>PSP Loader Program</u> - A PDP 11/45 program to transfer a PSP binary program from a PDP 11/45 source device to the PSP program and data memory. The source device may be the card reader, disk file, Mag tape, Dec tape, or paper tape unit.
2. PSPODT	<u>PSP On Line Debugging Technique Program</u> - A PDP 11/45 used to debug PSP programs. Single or multiple program or data memory locations in the PSP may be changed by specifying the address and new contents at the PDP 11/45 TTY (teletype or Dec Writer) input device. The old contents may be destroyed or remembered and subsequently restored by the PDP 11/45. The contents of PSP program or data memory locations may be examined via the PSP front panel light or printed on the TTY by specifying the locations to be examined at the PDP 11/45 TTY.
3. PSPMDP	<u>PSP Memory Dump Program</u> - A PDP 11/45 program to "dump" the contents of the PSP program or data memory, listing the contents on the PDP 11/45 TTY or line printer. This program is incorporated as an option in the PSPODT program described above.

TABLE 4-1. PSP - PDP 11/45 INTERFACE SOFTWARE (Cont.)

Group II - PSP to PDP 11/45 data transfers. This group operates with the PSP in the run mode. Each function requires matching pairs of programs: a program to transmit the data from the PSP and a receiving program in the PDP 11/45. Only the PSP routines are listed.

<u>PROGRAM</u>	<u>DESCRIPTION</u>
1. WRIT	A generalized PSP Macro to steer a record of data from the PSP to a PDP 11/45 output device.
2. RWND	A PSP Macro routine to request a rewind on a PDP 11/45 I/O device.
3. WEOF	A PSP Macro routine to write an end of file on a PDP 11/45 Output device.
4. BKSP	A PSP Macro routine to request a backspacing of records on a PDP 11/45 I/O device.
5. TYPE	A specialized PSP Macro within the generalized WRIT category to transfer a character string to the PDP 11/45 TTY device.
6. PNCH	A specialized PSP Macro within the generalized WRIT category to transfer a data string to the PDP 11/45 PUNCH device.

Group III - PDP 11/45 to PSP data transfers. The group operates with the PSP in the run mode. As in Group II each function requires matching pairs of programs: a PDP 11/45 program to transmit the data and a PSP routine to accept the data. Only the PSP routines are listed.

<u>PROGRAM</u>	<u>DESCRIPTION</u>
1. READ	A PSP generalized Macro to request a data transfer from a PDP 11/45 I/O device.
2. RDCH	A PSP Macro to read a character input at the PDP 11/45 teletype.
3. RDLN	A PSP Macro to read a line of characters input at the PDP 11/45 teletype.
4. RPTP	A PSP Macro to read data from a paper tape unit.

#### 4.1.3.3 USE OF THE COMBINED SOFTWARE

The user's task, as a result of these utility programs, will be reduced to the job of writing the PSP program using the standard text editing facilities of the PDP 11/45, or other standard means of creating the source program.

With the PSP program complete, the user will type a sequence of commands at the PDP 11/45 teletype such as:

RUN	PAP	(assembler PSP program)
RUN	PSPLDR	(load program to PSP)
RUN	PSPINT	(run interface program)

At this point the user will be ready to run the PSP program with the devices of the PDP 11/45 available as requested by the PSP program.

#### 4.2 PSP DIAGNOSTIC PROGRAMS

A complete set of diagnostic software has been developed for the PSP to verify that the PSP is working properly and/or aid in isolating a hardware problem in the event of a component failure. The programs available are:

- a. Data memory test program
- b. Program memory test program
- c. Register file test program
- d. Instruction test program
- e. Multiply test program

These programs are provided with the PSP as deliverable items. Since the PSP is a programmable signal processor, additional test programs to check out special I/O equipment, real-time counters, etc. can be easily developed.

The program listings are inclosed in Appendix 5.

#### 4.2.1 PSP Register File Test Program

4.2.1.1 Objective - The PSP register file test program tests the PSP to determine whether data can be stored and retrieved properly from the 32 index registers.

4.2.1.2 Method - The program follows the same procedure used for the PSP data memory test which is described in subsection 4.2.2. Although the objective, method and flow chart of the two programs are identical, the programs themselves reflect the differences in programming technique required to address the register locations. Because of these similarities, no flow chart is provided for the register file test.

4.2.1.3 Program Timing - Because there are only 32 registers, the program cycles very rapidly, continuously repeating the test unless a failure occurs.

4.2.1.4 Factors Not Tested - This test is designed to test the storage cells in the register file only. It is not intended to check the register file arithmetic unit which is tested elsewhere.

#### 4.2.2 PSP Data Memory Test Program

4.2.2.1 Objective - The data memory test program tests the PSP to determine whether data can be stored and retrieved properly from the data memory.

4.2.2.2 Method - The program tests the data memory by storing a background pattern of ZEROS in all data memory locations. A test word with a ONE in a single-bit position is then stored in the test location. The entire data memory is checked to determine whether the ONE and the ZEROS were stored correctly.

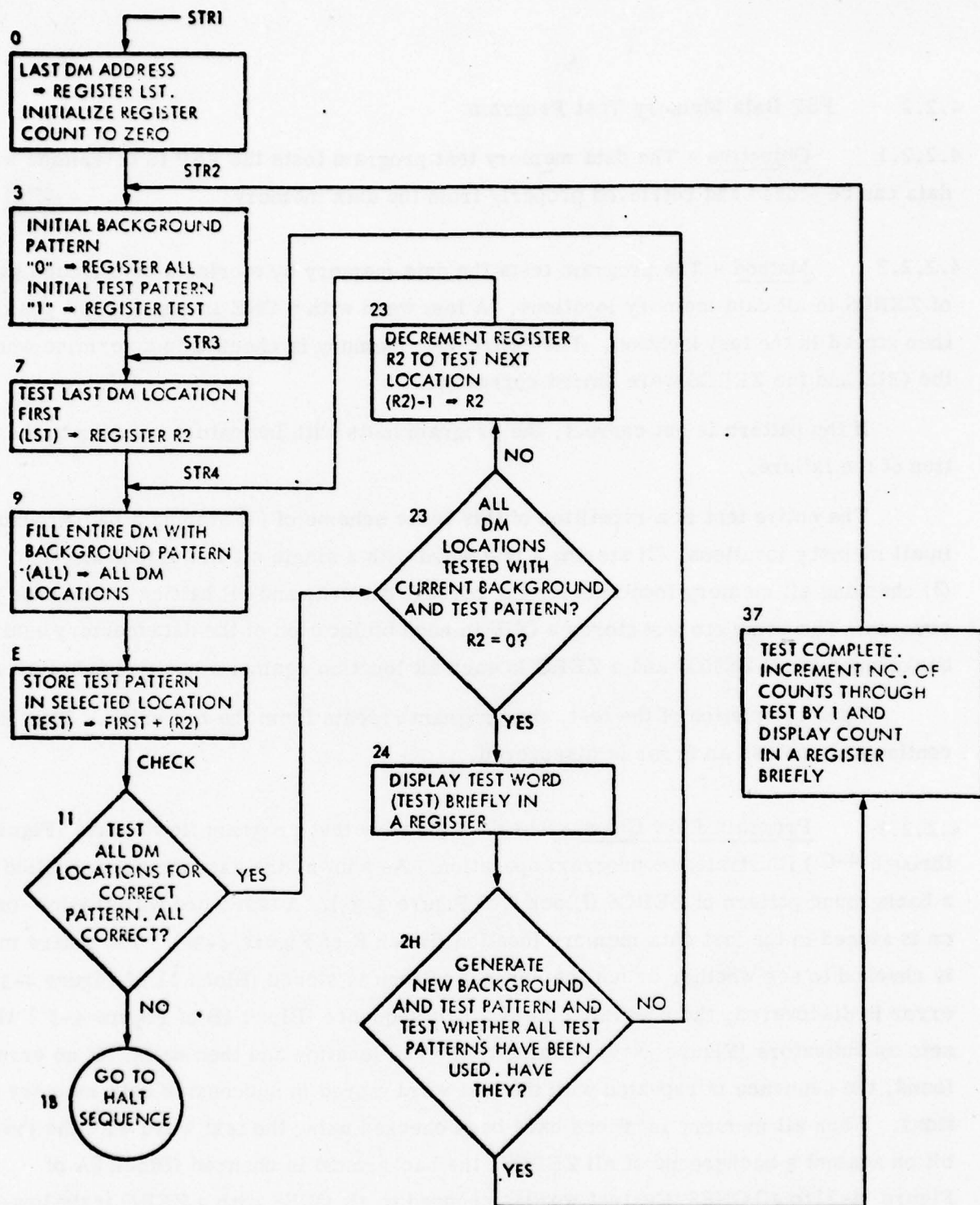
If the pattern is not correct, the program halts with indicators pointing to the location of the failure.

The entire test is a repetition of this basic scheme of (1) storing a background pattern in all memory locations, (2) storing a test word with a single bit in the test location, (3) checking all memory locations for the correct pattern, and (4) halting if an error is discovered. The complete test stores a ONE in each bit location of the data memory against a background of all ZEROS and a ZERO in each bit location against a background of all ONES.

Upon completion of the test, the program repeats from the beginning. Thus it runs continuously unless an error is discovered.

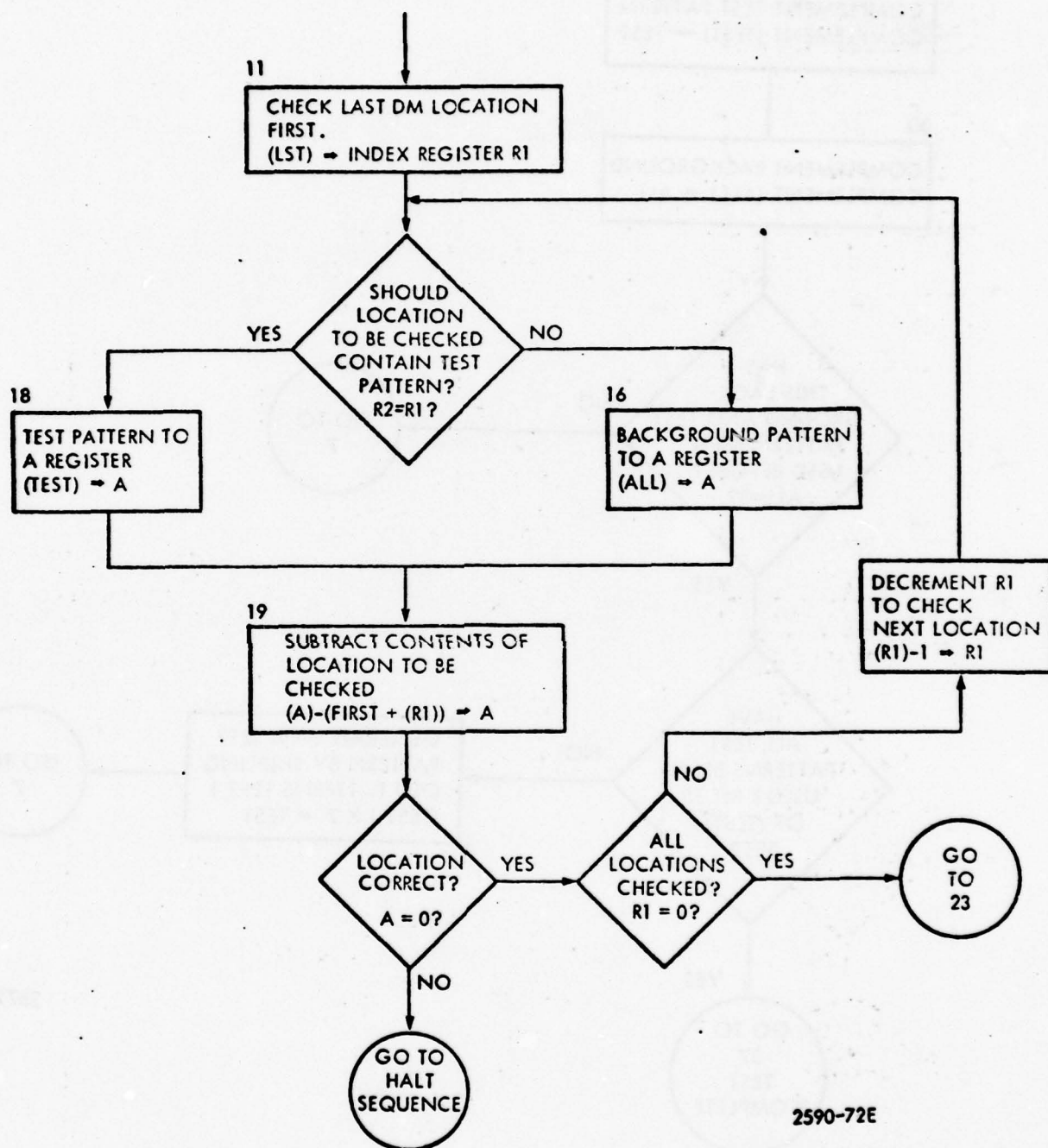
4.2.2.3 Program Flow Chart - The data memory test program flow charts (Figures 4-3 through 4-6 ) illustrate the program operation. As shown, the data memory is filled with a background pattern of ZEROS (Block 9 of Figure 4-3 ). A test word with the low-order bit on is stored in the last data memory location (Block E of Figure 4-3 ). The entire memory is checked to see whether or not the correct pattern is stored (Block 11 of Figure 4-3 ). If an error is discovered, the program enters a halt sequence (Block 1B of Figure 4-3 ) which sets up indicators (Figure 4-6) pointing to the bad location and then halts. If no error is found, the sequence is repeated with the test word stored in successive data memory locations. When all memory locations have been checked using the test word with the low-order bit on against a background of all ZEROS, the background is changed (Block 2A of Figure 4-3) to all ONES, the test word is changed to all ONES with a ZERO in the low-order bit location and the test sequence is repeated.

After all memory locations are tested with the second pattern of a background of ONES and a test word of ONES with a ZERO in the low-order position, the background and test word are again complemented, and the test word is shifted left one bit to move the bit which is on



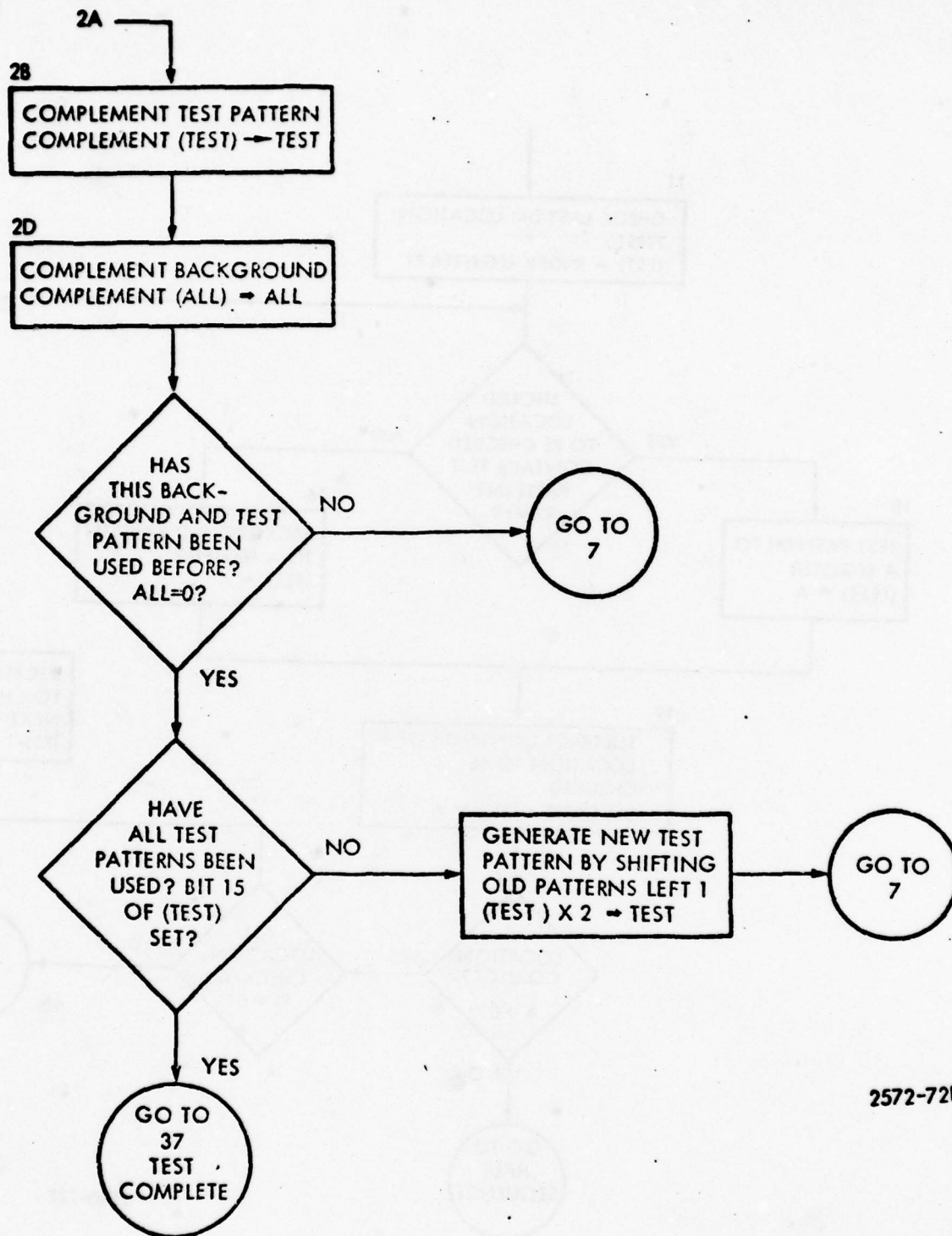
2589-72E

Figure 4-3 Data Memory Test Program Flow Chart



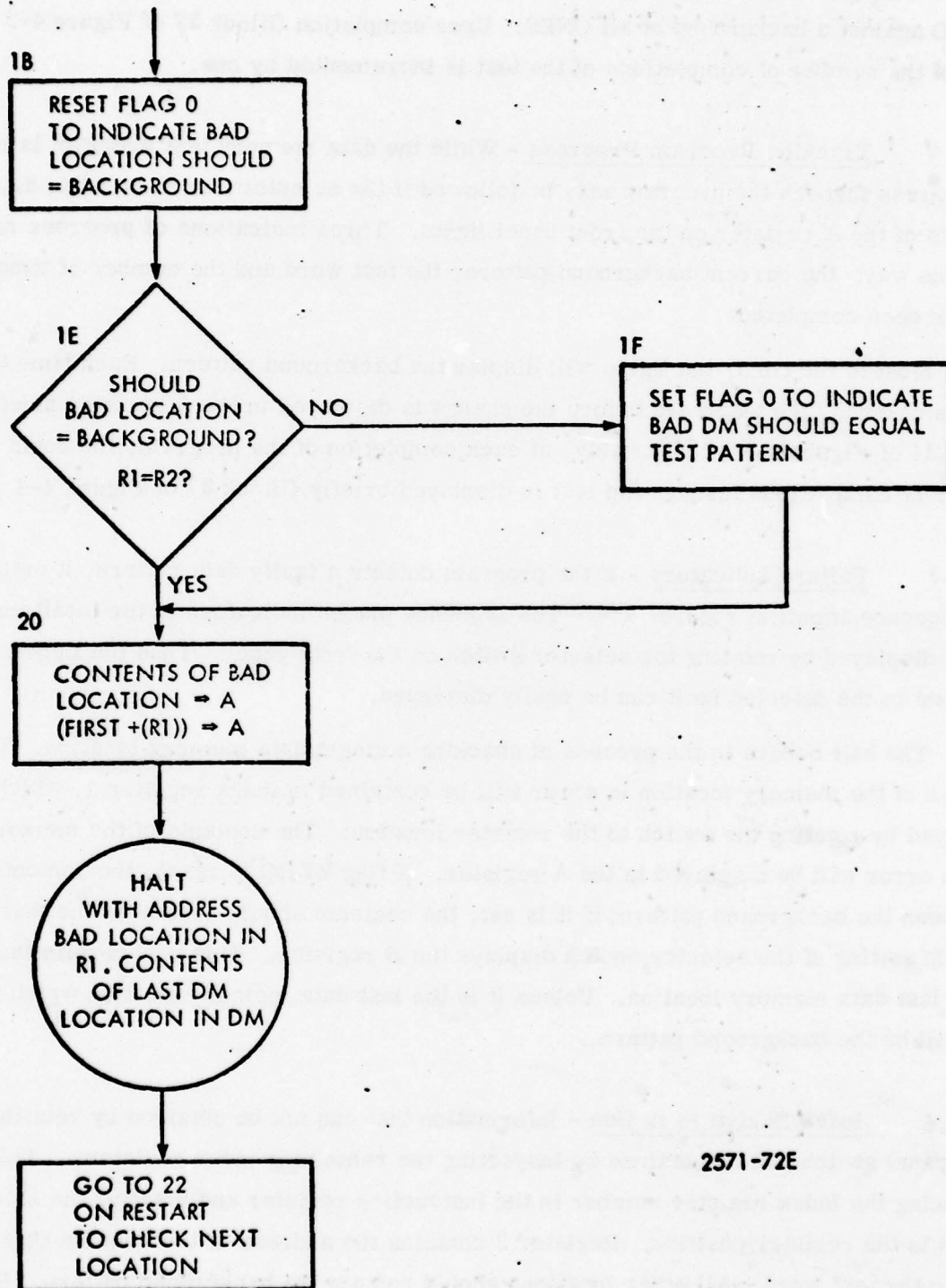
2590-72E

Figure 4-4 Data Memory Test Program Flow Chart -  
Test of All DM Locations for Correct Patterns



2572-72E

Figure 4-5 Data Memory Test Program Flow Chart -  
Generation of New Background and Test Patterns



2571-72E

Figure 4-6 Data Memory Test Program Flow Chart - Halt Sequence

in the test word to the next bit location. This sequence is repeated until all bit locations of the data memory have been tested with a ONE against a background of all ZEROS, and with a ZERO against a background of all ONES. Upon completion (Block 37 of Figure 4-3 ), a count of the number of completions of the test is incremented by one.

**4.2.2.4    Tracking Program Progress** - While the data memory test program is running, its progress through the program may be followed if the selector switch is set to display the contents of the A register on the front panel lights. Three indications of progress may be seen this way: the current background pattern, the test word and the number of times the test has been completed.

Most of the time, the lights will display the background pattern. Each time the test word is changed, the test word before the change is displayed in the A register briefly (Block 24 of Figure 4-3 ). Similarly, at each completion of the program, the count of the number of completions through the test is displayed briefly (Block 37 of Figure 4-3 ).

**4.2.2.3    Failure Indicators** - If the program detects a faulty data pattern, it enters the halt sequence shown in Figure 4-6. The sequence places indicators in the locations that can be displayed by rotating the selector switch on the front panel. Thus the information pertinent to the detected fault can be easily displayed.

The halt occurs in the process of checking a single data memory location. The address of the memory location in error will be contained in index register 1, which can be displayed by rotating the switch to the register location. The contents of the memory location in error will be displayed in the A register. If flag ZERO is reset, the contents should have been the background pattern; if it is set, the contents should have been the test pattern. The DM setting of the selector switch displays the B register. This will contain the contents of the last data memory location. Unless it is the last data memory location which failed, this will be the background pattern.

**4.2.2.6    Index Registers in Use** - Information that can not be obtained by rotating the front panel switch can be obtained by inspecting the remaining index registers. This is done by placing the index register number in the instruction register and rotating the selector switch to the register position. Register 2 contains the address of the location that should contain the test word. All other locations should contain the background pattern. Register 3

I  
contains the background pattern, while Register 4 contains the test word. Register 5 contains the address of the last data memory location. Register 6 contains the count of the number of successful completions of the test.

4.2.2.1 Memory Size Variations - The program has been written to check a data memory with 3072 locations. For different memory sizes, this may be easily changed. The first instruction in the program is a load register with literal instruction that places the number of locations minus 1 in register 5. By changing this one instruction the program will check data memories of any size. The program checks all locations starting with address ZERO. In the event that the starting address is to be changed, locations C, F, 19, and 20 must be changed. These are the locations that use the first data memory location as location ZERO. These should be changed to refer to the first location desired.

4.2.2.2 Program Timing - The time to run a complete test varies roughly as the square of the memory size, since the main test loop contains roughly  $N$  operations, where  $N$  is the memory size, and must be repeated  $N$  times. For a data memory with 1K locations, the program takes approximately two minutes, whereas for a 12K memory it takes about six hours.

Factors Not Tested - The data memory test program tests the individual storage locations, address decoding, card selection, and data paths. It does not test the various addressing modes such as indirect addressing. This is left to another program.

#### 4.2.3 Instruction Test Program

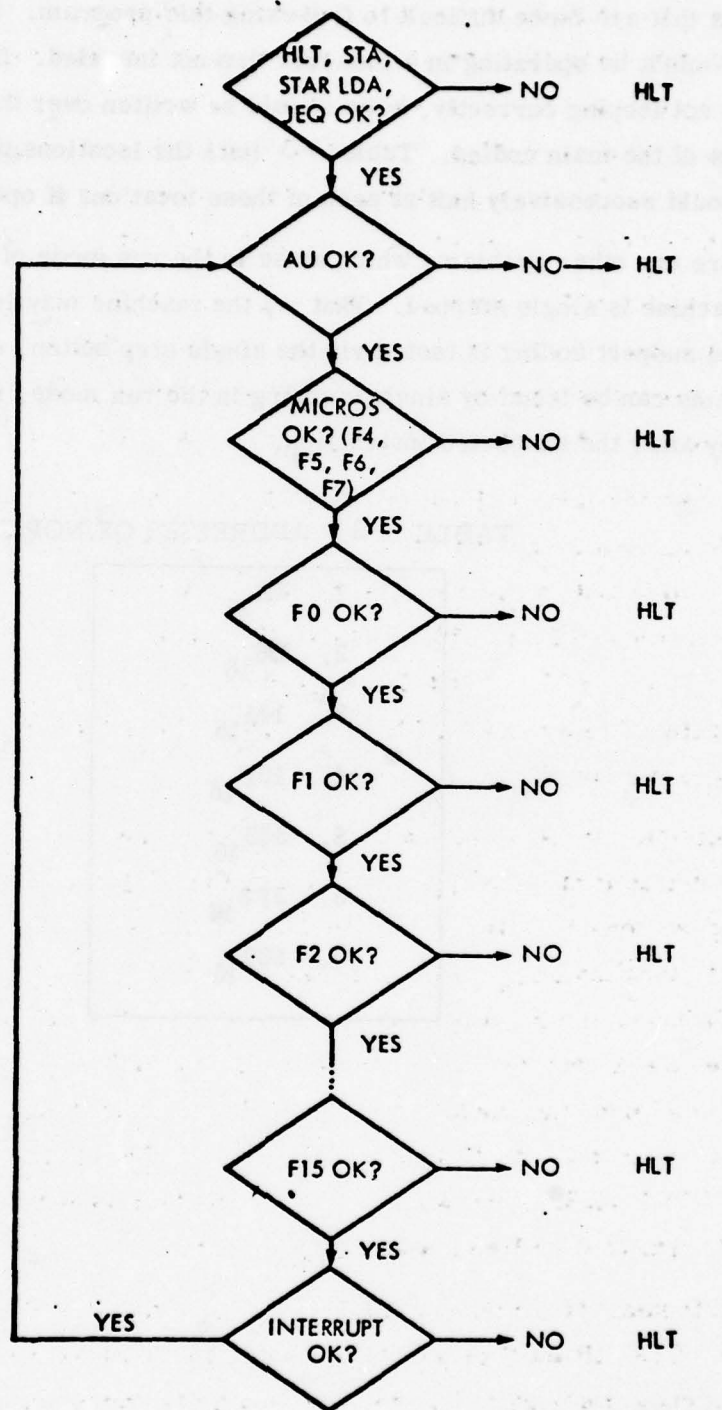
An attempt was made in the development of the instruction testing routine to keep in mind and check the particular hardware implementation chosen for the PSP instructions. That is, the program is more than a simple instruction verifier but something less than an exhaustive machine checking algorithm. Each instruction was pushed hard but not necessarily through all possible execution cases. For each instruction a small set of cases was chosen which would check the most likely set of single and some multiple faults.

The operation of the program is quite simple. If it runs and seems to be looping correctly the machine is apparently executing all of its instructions. If it halts (after an initial halt test) there is a problem. The location of that halt will point to the most likely set of instructions which failed. The program is flow charted in Figure 4-7.

The program begins in location STRT ( $D_{16}$ ). Before loading the program it is assumed that PM clear has been pushed and the A register is thus cleared. First, load, store, register store, JEQ and halt is checked by hitting the run switch. If halt worked, the program should stop at location  $18_{16}$  and the PC will read  $19_{16}$ . The operator should verify that  $A = \text{EMPTY} = R4 = \text{FFFF}_{16}$  and  $R1 = R2 = R3 = RM1 = RM2 = 0$  before proceeding and A should be restored to its full condition. If, after starting the program again at location  $19_{16}$ , the program halts, there is a problem.

For example, suppose the machine halts, and there is an  $A3_{16}$  in the PC. The halt has been executed in location  $A2_{16}$ , which means the JEQ in location  $A1_{16}$  failed. If the A register is indeed non-zero, then hand checking should begin at the beginning of this particular test, location  $9E_{16}$ . First, the operator should single instruct LDAL FFFF and check to see if indeed the A register is set to  $\text{FFFF}_{16}$ . ADDL 0 then sets the B register to zero and leaves the A register full. Finally .INSTRT 44622 takes as the A input to the AU the number 0 and forms the Boolean product of the A register ( $\text{FFFF}_{16}$ ) and B register ( $0000_{16}$ ), which is used as the B input to the AU, and adds the two inputs. If the result is non-zero, .INSTRT 44622 is not working. Perhaps the Boolean product is not zero, or perhaps the A input to the AU is not being set to zero.

Note that some of the tests can fail by jumping to halt, which is outside of the main line of coding. The halt location will still point to the instructions which failed. Suppose the machine stops at location  $546_{16}$ , the halt would have been executed at location  $545_{16}$  and the comment field of that halt tells us that we came from line  $232_{16}$  in the coding. The operator would then single instruct through the appropriate coding to see why it did jump on bit 8 set when it should have been reset.



2582-72E

Figure 4.7 Instruction Test Program Flow Chart

So much for the obvious failures. There are a number of problems which lead to subtle faults that are more difficult to find using this program. In fact, the program may not halt but might be operating in a loop that was not intended. If it is suspected that the program is not looping correctly, halts should be written over the NOP's which separate large blocks of the main coding. Table 4-3 lists the locations to insert halts. The machine should successively halt at each of these locations if operating properly.

There are other problems which arise in the run mode of operation but are eliminated when the machine is single stepped. That is, the machine may indeed halt at some time, but when the suspect coding is tested via the single step button, everything appears in order. Such problems can be found by single stepping in the run mode, i.e., by writing a halt immediately after the suspected instruction.

TABLE 4-3 . ADDRESSES OF NOP'S

1.	2E <sub>16</sub>
2.	D6 <sub>16</sub>
3.	144 <sub>16</sub>
4.	10F <sub>16</sub>
5.	353 <sub>16</sub>
6.	3F3 <sub>16</sub>
7.	507 <sub>16</sub>

#### 4.2.4 Program Memory - Memory Address Test

The intent of this program is to test the address wiring of the PSP's program memory. To test the addressing of memory the program must write data into memory and read it back for comparison to be sure the data was stored and returned from the proper location.

Since the instructions stored in program memory cannot write or read from program memory, the card reader is used to program the PSP's console to effect this test.

The test used is a 3-pass test. The first pass addresses each location of a 1K octal sector of memory storing that location's address in the memory address position of its 32-bit word.

The second pass programs the PSP's console: to read a 1K octal sector of program memory; to compare the memory address position of each location with its address; and to stop the card reader when the location contents are not equal to the location address.

The third pass is used to read a 1K sector of memory and determine if data had inadvertently been written there during the writing of another 1K sector.

Figures 4-8 , 4-9 and 4-10 are flow charts of PASS 1, 2 and 3 operation.

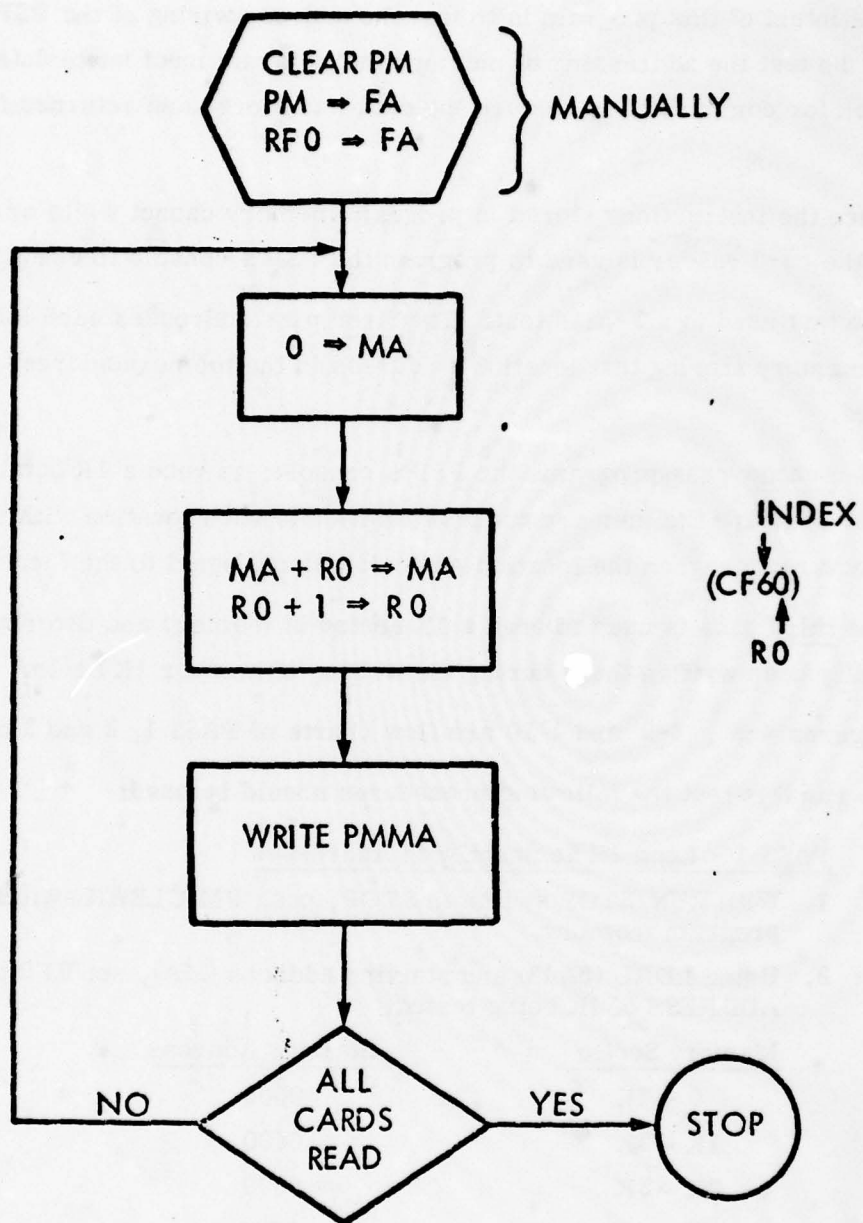
To run this test the following procedures should be used:

a. PASS 1 - Load 1K Segment with Addresses

1. With RUN/STOP switch in STOP, push PM CLEAR switch to clear program memory.
2. Using LDRL (3A40) and starting address (MA), set RF0 to FIRST ADDRESS of 1K being tested.

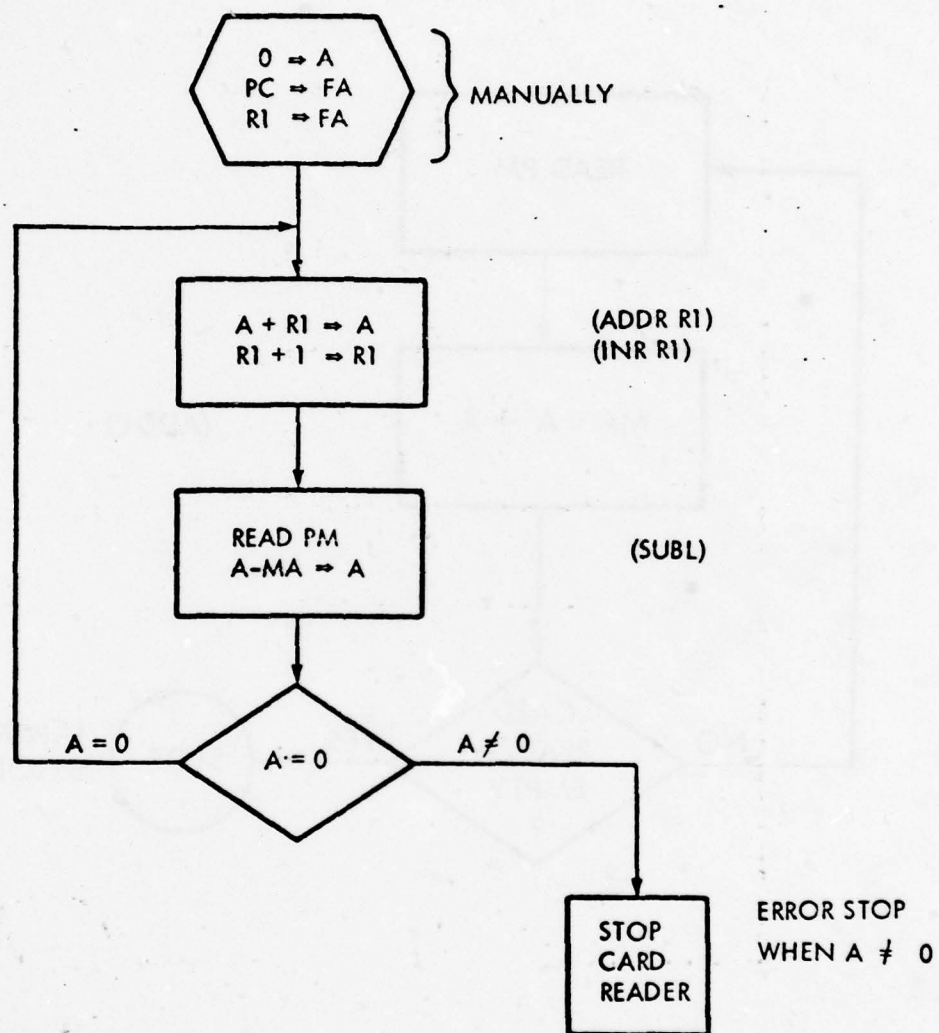
<u>Memory Sector</u>	<u>Starting Address</u>
0 - 1K	0000
1K - 2K	0400
2K - 3K	0800
3K - 4K	0C00

3. Set PC to FIRST ADDRESS; JMP (8000) to starting address.
4. Place PASS 1 card deck on card reader and start the reader. The MA DISPLAY REGISTER should step through the 1K locations of program memory as they are loaded.



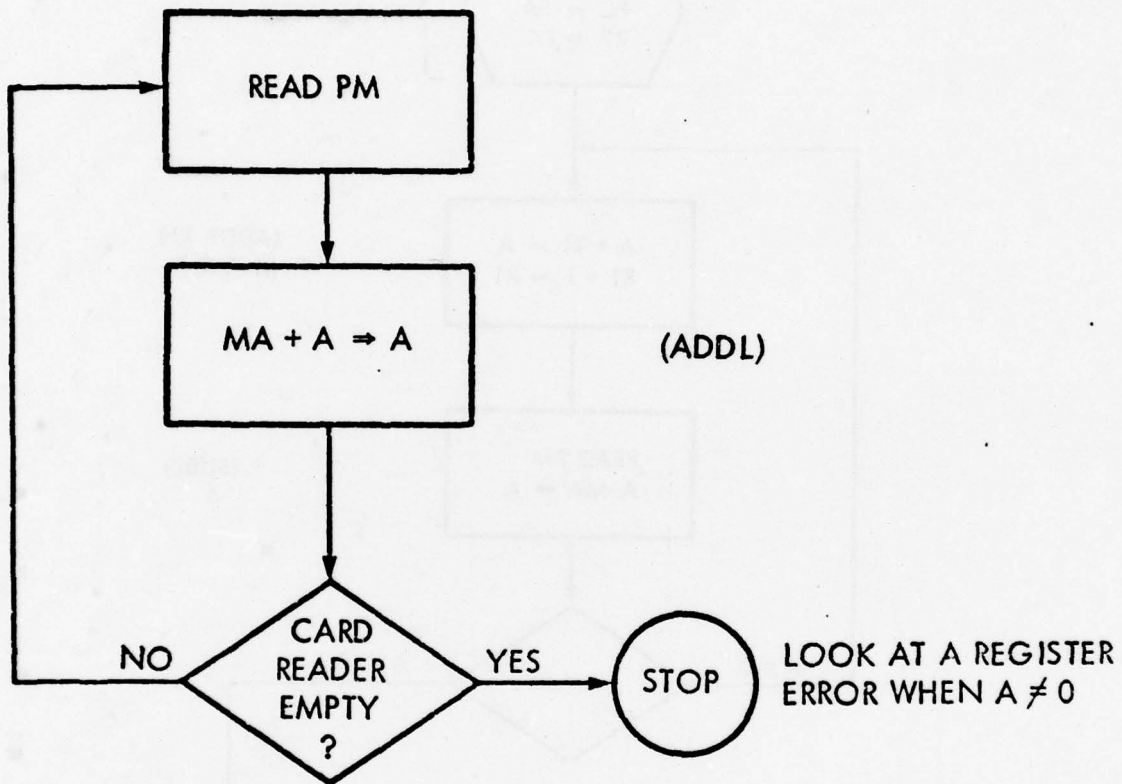
2601-72E

Figure 4-8 PASS 1 Flow Chart



2581-72E

Figure 4-9 PASS 2 Flow Chart



2575-72E

Figure 4-10 PASS 3 Flow Chart

**b. PASS 2 - Check 1K for Valid Address Storage**

1. Using steps 2 and 3 in PASS 1 procedure, set RF1 and PC to the FIRST ADDRESS of 1K being tested. For step 2 use LDRL (3A41) to load RF1.
2. Using CLA (4343), clear the A register.
3. Place PASS 2 deck on card reader and start reader. The card reader should stop with all cards read and the A register equal to 0000 (HEX), indicating that the 1K segment has been read and does contain the information as written during PASS 1.

**c. PASS 3 - Test Other 1K Segments for all Zeros**

1. Set RF0 and PC to the FIRST ADDRESS of a 1K segment other than the segment containing its addresses; similar to steps 2 and 3 in PASS 1.
2. Using CLA (4343), clear the A register.
3. Place PASS 3 on card reader and start card reader. During this PASS, the A register holds the sum of error words found.

A register equals 0000 (HEX) for segment holding 0000 (HEX) in all locations.

**4.2.5 PSP Multiply Test Program**

The intent of this program is to test the PSP's arithmetic unit and associated logic using the fractional multiply instruction. This instruction, MLNR, multiplies the contents of a register file by the contents of the A register, leaving the results in the A register.

The program using this instruction multiplies all combinations of a 16-bit word found in register file 0 (R0) by all combinations of another 16 bit word found in register file 1 (R1).

Each set of words held in R0 and R1 are multiplied twice. During the first multiply R0 contains A and R1 contains B. During the second multiply R0 contains B and R1 contains A. Upon completion, the results of both multiplies are compared for equality.

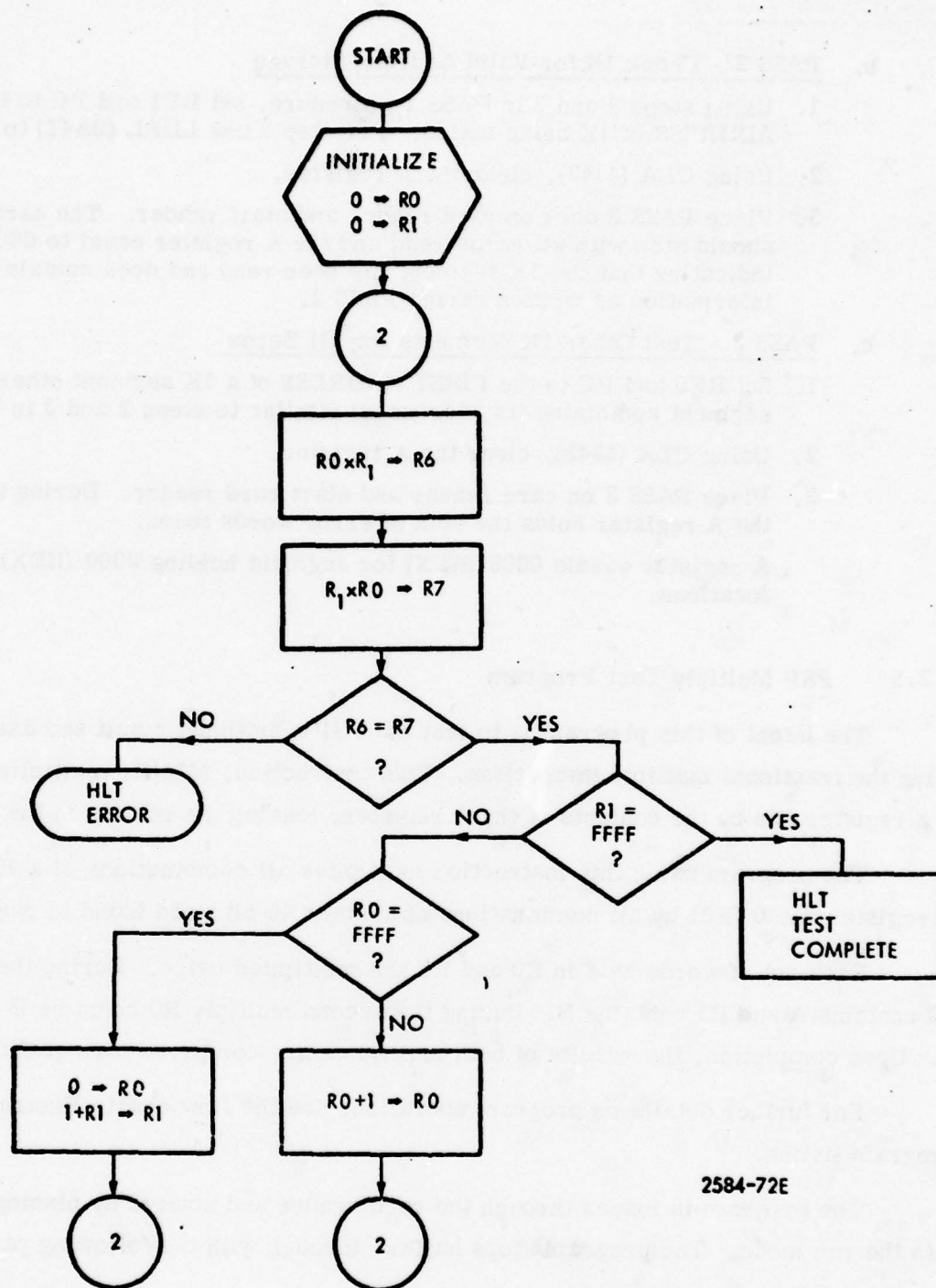
For further details on program operation, see the flow chart, Figure 3-36 and the program listing.

The program is loaded through the card reader and started by placing the run switch into the run mode. The program stops halfway through with the following particulars:

ASN = 1

RF0 = 8000

PC = B



2584-72E

Figure 4-11 PSP Multiply Test Program Flow Chart

When this happens, to continue, position the run switch down momentarily and then back to run. The program will run for approximately another two hours to completion, when it will stop with PC = 11.

Other halts occurring at location OA are due to multiply errors.

APPENDIX A

DECODER TEST ROUTINES

# DECODING EXAMPLE (25,16) CODE

BIT No.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1	X	25		invert
2	2		24		forced
3	3		21		forced
4	4		19		forced
5	5		17		forced
6	6	X	23		invert ; force bit 1
7	7		22		force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		flag bit 6, force bit 5
11	11	X	15		invert
12	12		14		force bit 11
13	13		13		
14	14		12		
15	15		11		flag bit 11
16	16	X	10		invert by forcing
17	17		9		force bit 16
18	18		8		
19	19		7		
20	20		6		flag bit 16
21	21		5		forced
22	22		4		
23	23		3		
24	24		2		
25	25		1		

4 bits in error, all errors in one column,  
all errors are on lowest ranked bits in four rows,  
no errors after decoding

# DECODING EXAMPLE (25,16) CODE

BIT No.	RANK	ERRORS	ORDER of DECODING	FINAL ERROR STATUS	NOTES
1	1		25		forced
2	2		24		forced
3	3		21		forced
4	4		19		forced
5	5		17		forced
6	6		23		forced
7	7		22		force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		force bit 5
11	11		15		forced
12	12		14		force bit 11
13	13		13		
14	14		12		
15	15		11		
16	16	X	10		flagged; forced, insert
17	17		9		force bit 16
18	18		8		
19	19		7		
20	20		6		flag bit 16
21	21	X	5		flagged; force, insert
22	22		4		force bit 21
23	23		3		
24	24		2		
25	25		1		flag bit 21

2 bits in error, both of high rank,  
both are lowest ranked bit in a row,  
all bits decoded correctly

# DECODING EXAMPLE (25,16) CODE

BIT N°	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1	X	25		inverted in step 3
2	2		21		forced
3	3	X	24		inverted in step 3
4	4		17		forced
5	5		15		forced
6	6		20		forced
7	7		19		force bits 6 and 2.
8	8		18		
9	9		16		force bit 4
10	10		14		force bit 5
11	11		13		forced
12	12		12		force bit 11
13	13		11		
14	14		10		
15	15		9		
16	16		8		forced
17	17		7		force bit 16
18	18		6		
19	19		5		
20	20		4		
21	21	X	23		inverted in step 3; force bit 1
22	22		3		flag bit 21
23	23		22		flagged; made in step 3; force 3, 21, 24
24	24		2		
25	25		1		flag 21

3 bits in error, one of high rank (lowest ranked bit in row) all bits corrected

# DECODING EXAMPLE (25,16) CODE

BIT NO.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1	x	25	x	forced
2	2		24		forced
3	3	x	21	x	forced
4	4		19		forced
5	5		17		forced
6	6		23		free bit 1
7	7		22		free bits 6 and 2
8	8		20		free bit 3
9	9		18		forced bit 4
10	10		16		free bit 5
11	11		15		forced
12	12		14		free bit 11
13	13		13		
14	14		12		
15	15		11		
16	16		10		forced
17	17		9		free bit 16
18	18		8		
19	19		7		
20	20		6		
21	21		5	x	invert
22	22		4		free bit 21
23	23	x	3	x	
24	24		2		
25	25		1		flag bit 21

3 bits in error, one of high rank (not lowest ranked bit in row)  
4 errors after decoding

# DECODING EXAMPLE (25,16) CODE ✓

BIT No.	RANK	ERRORS	ORDER of DECODING	FINAL ERROR STATUS	NOTES
1	1		25	X	invited in step 3
2	2		24	X	invited in step 3
3	3		19		forced
4	4		17		forced
5	5		15		forced
6	6		21		
7	7		20		force 6
8	8		18		force 3
9	9		16		force 4
10	10		14		force 5
11	11		13		forced
12	12	X	12	X	force 11
13	13	X	11	X	
14	14		10		
15	15		9		
16	16		23	X	flagged; invited in step 3, force 1
17	17		22		flagged; decoded in step 3; force 16,
18	18	X	8	X	
19	19		7		
20	20		6		flag 16
21	21		5		forced
22	22		4		force 21
23	23		3		
24	24		2		
25	25		1		

3 bits in error, all of high rank  
6 errors after decoding

# DECODING EXAMPLE (25,16) CODE ✓

BIT No.	RANK	ERRORS	ORDER of DECODING	FINAL ERROR STATUS	NOTES
1	1		25		forced
2	2		24		forced
3	3		21		forced
4	4		19		forced
5	5		17		forced
6	6		23		force bit 1
7	7		22		force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		force bit 5
11	11	X	15		flagged; inverted - forced
12	12		14		force bit 11
	13		13		
14	14		12		
15	15		11		flag bit 11
16	16		10		forced
17	17		9		force bit 16
18	18		8		
19	19		7		
20	20		6		
21	21		5		forced
22	22		4		force bit 21
23	23		3		
24	24		2		
25	25		1		

1 high rank bit in error;  
but it is the lowest ranked bit in a row,  
all bits corrected

# DECODING EXAMPLE (25,16) CODE ✓

BIT No.	RANK	ERRORS	ORDER of DECODING	FINAL ERROR STATUS	NOTES
1	1	X	25		invert
2	2	X	24		invert
3	3	X	21		invert
4	4	X	19		invert
5	5		17		forced
6	6		23		forced ; force bit 1
7	7		22		force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		force bit 5
11	11		15		forced
12	12		14		force bit 11
13	13		13		
14	14		12		
15	15		11		
16	16		10		forced
17	17		9		force bit 16
18	18		8		
19	19		7		
20	20		6		
21	21		5		forced
22	22		4		force bit 21
23	23		3		
24	24		2		
25	25		1		

4 bits in error, all of low rank, all in one row, no errors after decoding

# DECODING EXAMPLE (25,16) CODE

BIT No.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1		25		
2	2	✓	24	X ✓	invert
3	3	✓	21	X ✓	invert
4	4	✓	19	X ✓	invert
5	5		17	X	invert
6	6	✓	23	X ✓	invert
7	7	X	22	X ✓	force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		force bit 5
11	11		15	X ✓	invert
12	12		14		force bit 11
13	13	X	13	X ✓	
14	14		12		
15	15		11		
16	16	✗	10	X ✓	invert
17	17		9		force bit 16
18	18		8		
19	19	X	7	X ✓	
20	20		6		
21	21		5	X ✓	invert
22	22		4		force bit 21
23	23		3		
24	24		2		
25	25	X	1	X	

all bits decoded in step 3 of algorithm  
 4 errors of high rank on diagonal  
 12 errors after decoding

# DECODING EXAMPLE (25,16) CODE ✓

BIT N°.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1		25	x	invert
2	2		24	x	invert
3	3		21	x	invert
4	4		19	x	invert
5	5		17		forced
6	6		23		forced ; force bit 1
7	7		22		force bits 6 and 2
8	8		20		force 3
9	9		18		force 4
10	10		16		force 5
11	11		15		forced
12	12		14		force 11
13	13		13		
14	14		12		
15	15		11		
16	16		10		forced
17	17		9		force 16
18	18		8		
19	19		7		
20	20		6		
21	21	X	5	x	forced
22	22	X	4	x	force 21
23	23	X	3	x	
24	24	X	2	x	
25	25		1		

4 bits in error, all errors are in one row,  
all errors are on highest ranked bits in 4 columns,  
8 errors after decoding

# DECODING EXAMPLE (23,16) CODE

BIT N°.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1		25	X	invert
2	2		24		forced
3	3		21		forced
4	4		19		forced
5	5	X	17	X	forced
6	6		23	X	invert, force 1
7	7		22		force bits 6 and 2
8	8		20		force 3
9	9		18		force 4
10	10	X	16	X	flag 6, <del>and</del> force 5
11	11		15	X	invert
12	12		14		force 11
13	13		13		
14	14		12		
15	15	X	11	X	flag 11
16	16		10	X	invert
17	17		9		force 16
18	18		8		
19	19		7		
20	20	X	6	X	flag 16
21	21		5		forced
22	22		4		force 21
23	23		3		
24	24		2		
25	25		1		

4 bits in error, all errors in one column,  
all errors are on highest ranked bits in four rows,  
8 errors after decoding

# DECODING EXAMPLE (25,16) CODE (C)

BIT NO.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1		25	X	forced, invert
2	2		24	X	forced, invert
3	3		21		forced
4	4		19		forced
5	5		17		forced
6	6		23		forced, force bit 1
7	7		22		force bits 6 and 2
8	8		20		force bit 3
9	9		18		force bit 4
10	10		16		force bit 5
11	11		15		forced
12	12		14		force bit 11
	13		13		
14	14		12		
15	15		11		
16	16	X	10		flagged; forced, invert
17	17		9		force bit 16 FLAG 17
18	18		8		
19	19		7		
20	20		6		flag bit 16
21	21	X	5	X	forced
22	22	X	4	X	force bit 21
23	23		3		
24	24		2		
25	25		1		

3 bits in error, all of high weight, 2 in one row  
4 bits in error after decoding

# DECODING EXAMPLE (25,16) CODE

BIT N°.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1	X	25		inverted in step 3
2	2		16		forced
3	3		24	X ✓	inverted in step 3
4	4		14		forced
5	5		21	X	flagged; inverted in step 3
6	6		23	X ✓	flagged; inverted in step 3; force 1
7	7		15		force bit 2, flag 5
8	8	X	22	X ✓	flagged; decoded in step 3; force 6, 3
9	9		13		flag bit 6; force bit 4
10	10		20		flagged; decoded in step 3; force 5
11	11		19	X ✓	flagged; inverted in step 3
12	12		12		
13	13		18		flagged; decoded in step 3; force 11
14	14		11		flag bit 11
15	15	X	17	X	flagged; decoded in step 3
16	16		10		forced
17	17		9		force bit 16
18	18		8		
19	19		7		
20	20		6		
21	21		5		forced
22	22		4		force bit 21
23	23		3		
24	24		2		
25	25		1		flag bit 5

3 bits in error, 2 of high rank.  
6 errors after decoding

# DECODING EXAMPLE (25,16) CODE X

BIT N°.	RANK	ERRORS	ORDER OF DECODING	FINAL ERROR STATUS	NOTES
1	1		25	X	inverted in step 3
2	2		17		forced
3	3		15		forced
4	4		13		forced
5	5		24	X	flagged; inverted in step 3
6	6		23	X	flagged; inverted in step 3; force bit 1
7	7		16		force bit 2
8	8		14		force bit 3
9	9		12		flag bit 6; force bit 4
10	10	X	22	X	flag; decoded in step 3; force 6, 5
11	11		21	X	flagged; inverted in step 3
12	12		11		
13	13		10		
14	14		9		flag bit 11
15	15	X	20	X	flagged; decoded in step 3; force 11
16	16		19	X	flagged; inverted in step 3
17	17		8		
18	18		7		
19	19		6		flag 16
20	20	X	18	X	flagged; decoded in step 3; force 16
21	21		5		forced
22	22		4		force 21
23	23		3		
24	24		2		
25	25		1		flag 5

Bits 1, 5, 6, 10, 11, 15, 16, and 20 decoded in step 3  
 3 errors of high rank in one column  
 all errors are on highest ranked bits in 3 rows  
 7 errors after decoding

APPENDIX B

DIAGNOSTIC LISTINGS

# PSP DATA MEMORY TEST PROGRAM 5/1/72

1			'PARAMETERS	
2	000000000		FIRST=0	'FIRST DM ADDRESS
3	0000003FF		LAST=3071.	'LAST DM ADDRESS
4	000000FFF		ASTEP=4095.	
5	0000003ER		BSTEP=1000.	
6			'	
7			'REGISTER ASSIGNMENTS	
8	000000001		R1=1	
9	000000002		R2=2	'REG 2 HOLDS TEST LOC
10	000000003		ALL=3	'REG 3 HOLDS BACKGND
11	000000004		TEST=4	'REG 4 HOLDS TEST WD
12	000000005		LST=5	'REG 5 HOLDS LAST DM LOC
13	000000006		COUNT=6	
14	000000007		R7=7	
15	000000008		R8=8.	
16			'	
17			'PROGRAM	
18	00000000G		.PRG 0	
19	0000 3A45 0BFF	STR1\$	LD,R LAST,LST	
20	0001 7A43 0000		LD,L 0	
21	0002 3AC6 FFFF		STAR COUNT	'INITIALIZE COUNT
22	0003 7A43 0000	STR2\$	LD,L 0	
23	0004 3AC3 FFFF		STAR ALL	'INITIALIZE BACKGND
24	0005 7A43 0001		LD,L 1	
25	0006 3AC4 FFFF		STAR TEST	'INITIALIZE TEST WD
26	0007 6A45 0000	STR3\$	LD,R LST	
27	0008 3AC2 FFFF		STAR R2	'START LAST MEM LOC.
28	0009 6A45 0000	STR4\$	LD,R LST	'FILL MEMORY
29	000A 3AC1 FFFF		STAR R1	'WITH
30	000B 6A43 0000		LD,R ALL	'BACKGND
31	000C FF21 0000		S FIRST,R1	'DATA.
32	000D 1141 000C		JR NZ,D.-1,R1	
33	000E 6A44 0000		LD,R TEST	'STORE TEST WD
34	000F FF22 0000		S FIRST,R2	'IN MEM LOC R2.
35	0010 4F43 0000	CHECK\$NOP		'(FOR PROG TEST HALT)
36				'(TO INSERT ERROR)
37				'(MANUALLY.)
38	0011 6A45 0000		LD,R LST	
39	0012 3AC1 FFFF		STAR R1	
40	0013 6A41 0000	CHK1\$	LD,R R1	'CHECK LOC R1
41	0014 6622 0000		SUB,R R2	'COMPARE LOC OF TEST WD
42	0015 2080 0018		JEQ GTST	'R1=R2
43	0016 6A43 0000		LD,R ALL	'NO, GET BACKGND
44	0017 8000 0019		J.-+2	
45	0018 6A44 0000	GTST\$	LD,R TEST	'YES, GET TEST WD
46	0019 D621 0000		SUB FIRST,R1	'MEM LOC R1 OK
47	001A 2080 0022		JEQ RTN	'YES, RETURN.
48	001B 00C0 0000		RF 0	'NO, GET RDY TO HLT.
49				'RESET FLAG TO
50				'INDICATE DM
51				'SHOULD=BACKGND.
52	001C 6A41 0000		LD,R R1	
53	001D 6622 0000		SUB,R R2	'SHOULD IT

54	001F	20A0	0020	JNE .+2	'YES, SKIP SF.
55	001F	08C0	0000	SF 0	'NO, SET FLAG.
56	0020	CA21	0000	LD FIRST,R1	'GET CONTENTS
57					'OF BAD MEM LOC
58					'FOR DISPLAY IN
59					'A REGISTER.
60	0021	0001	08FF	HLT LAST,R1	'HALT WITH
61					'BAD MEM LOC ADDRESS
62					'DISPLAYED
63					'IN RF AND LAST
64					'MEM LOC CONTENTS
65					'DISPLAYED
66					'IN DM.
67	0022	1141	0013	RTN\$ JRNZ,D CHK1,R1	'CHECK NEXT LOC
68	0023	1142	0009	JRNZ,D STR4,R2	'REPEAT TEST WD
69					'FOR ALL MEM LOC.
70	0024	6A44	0000	NPAT\$ LD,R TEST	'GET NEW PATTERN
71	0025	3A47	FFFF	LDRL 17777,R7	'BRIEF PAUSE
72	0026	3A48	000F	LDRL 17,R8	
73	0027	1147	0027	JRNZ,D .,R7	'TO DISPLAY
74	0028	3A47	FFFF	LDRL 17777,R7	'TEST WD.
75	0029	1148	0027	JRNZ,D .-2,R8	
76	002A	4F43	0000	NOP	'(FOR PROG TEST HALT)
77	002B	4043	0000	CPL	'COMPLEMENT TEST WD
78	002C	3AC4	FFFF	STAR TEST	
79	002D	6A43	0000	LD,R ALL	
80	002E	4F43	0000	NOP	'(FOR PROG TEST HALT)
81	002F	4043	0000	CPL	'COMPLEMENT BACKGND
82	0030	3AC3	FFFF	STAR ALL	
83	0031	20A0	0007	JNE STR3	'ALL=0
84	0032	6A44	0000	LD,R TEST	'YES
85	0033	2740	0037	JBS 17,DSPL	'ALL BITS TESTED
86	0034	4C03	0000	SL1	'NO, SHIFT TEST BIT 1
87	0035	3AC4	FFFF	STAR TEST	
88	0036	8000	0007	J STR3	'REPEAT FOR NEXT BIT
89				'DISPLAY ROUTINE	
90	0037	6A46	0000	DSPL\$ LD,R COUNT	'YES, TEST COMPLETE
91	0038	7903	0001	ADD,L 1	'INCREMENT COUNT
92	0039	3AC6	FFFF	STAR COUNT	
93	003A	3A47	0FFF	LDRL ASTEP,R7	
94	003B	3A48	03E8	LDRL BSTEP,R8	
95	003C	1147	003C	PAUS\$ JRNZ,D PAUS,R7	'PAUSE, DISPLAY
96	003D	3A47	0FFF	LDRL ASTEP,R7	'COUNT IN
97	003E	1148	003C	JRNZ,D PAUS,R8	'A REGISTER FOR
98					'THREE SECONDS
99	003F	8000	0003	J STR2	
100		00000000		.END STR1	

# CHECKSUMS

LEFT PROGRAM MEMORY= 001CD  
RIGHT PROGRAM MEMORY= 03D92  
DATA MEMORY= 00000

1		5/4/72		'REGISTER FILE TEST PROGRAM
2				'
3				'PROGRAM RUNS CONTINUOUSLY
4				'IF REGISTER FILE IS OK.
5				'DISPLAYING SLOWLY INCREASING
6				'NUMBER IN A REGISTER TO
7				'INDICATE THE NUMBER
8				'OF SUCCESSFUL COMPLETIONS
9				'OF TEST.
10				'
11				'PROGRAM HALTS IF ERROR
12				'IS FOUND.
13				'SELECT SWITCH MAY BE USED
14				'TO DISPLAY RELEVANT
15				'DATA AFTER THE HALT.
16				'PC-CONTAINS HALT LOCATION+1
17				'IR-R FIELD INDICATES BAD REG
18				'DM-REG WITH TEST WD
19				'A-CONTAINS TEST WD
20				'RF-CONTAINS BAD REG CONTENTS
21				'STATUS INDICATES IF
22				' FLAG 0=0, RF SHOULD=BACKGND
23				' FLAG 0=1, RF SHOULD=TEST WD
24				'
25				'
26		000000FFF		ASTEP=7777
27		000000200		ASTEP=1000
28		000000004		R4=4
29		000000005		R5=5
30		000000000		.PRNG 0
31				'START PROGRAM
32	0000	7A43 0000	STR1\$	LDAL 0
33	0001	FF00 0002		S COUNT 'INITIALIZE COUNT
34	0002	7A43 0000	STR2\$	LDAL 0
35	0003	FF00 0003		S ALL 'INITIALIZE BACKGND
36	0004	7A43 0001		LDAL 1
37	0005	FF00 0004		S TEST 'INITIALIZE TEST WD
38	0006	7A43 0000	STR3\$	LDAL 0
39	0007	FF00 0005		S RTEST 'START REG 0
40	0008	CAC0 0003	STR4\$	LD ALL
41	0009	3AC0 FFFF		STAR 0 'LOAD
42	000A	3AC1 FFFF		STAR 1 'ALL
43	000B	3AC2 FFFF		STAR 2 'REGISTERS
44	000C	3AC3 FFFF		STAR 3 'WITH
45	000D	3AC4 FFFF		STAR 4 'BACKGROUND
46	000E	3AC5 FFFF		STAR 5 'DATA.
47	000F	3AC6 FFFF		STAR 6
48	0010	3AC7 FFFF		STAR 7
49	0011	3AC8 FFFF		STAR 8.
50	0012	3AC9 FFFF		STAR 9.
51	0013	3ACA FFFF		STAR 10.
52	0014	3ACB FFFF		STAR 11.
53	0015	3ACC FFFF		STAR 12.

54	0016	3ACD FFFF	STAR 13.
55	0017	3ACE FFFF	STAR 14.
56	0018	3ACF FFFF	STAR 15.
57	0019	3AD0 FFFF	STAR 16.
58	001A	3AD1 FFFF	STAR 17.
59	001B	3AD2 FFFF	STAR 18.
60	001C	3AD3 FFFF	STAR 19.
61	001D	3AD4 FFFF	STAR 20.
62	001E	3AD5 FFFF	STAR 21.
63	001F	3AD6 FFFF	STAR 22.
64	0020	3AD7 FFFF	STAR 23.
65	0021	3AD8 FFFF	STAR 24.
66	0022	3AD9 FFFF	STAR 25.
67	0023	3ADA FFFF	STAR 26.
68	0024	3ADB FFFF	STAR 27.
69	0025	3ADC FFFF	STAR 28.
70	0026	3ADD FFFF	STAR 29.
71	0027	3ADE FFFF	STAR 30.
72	0028	3ADF FFFF	STAR 31.
73	0029	7A43 002F	LUAL FIRST 'CALCULATE JUMP
74	002A	5905 0003	ADD,2M RTEST 'LOCATION.
75	002B	FF00 0006	S WHICH
76	002C	CA00 0004	LU TEST 'GET TEST WD
77	002D	8080 0006	J WHICH
78	002E	3AC0 FFFF	FIRST\$STAR 0 'STORE
79	002F	8000 006E	J CHECK 'TEST
80	0030	3AC1 FFFF	STAR 1. 'PATTERN
81	0031	8000 006E	J CHECK 'IN
82	0032	3AC2 FFFF	STAR 2. 'SELECTED
83	0033	8000 006E	J CHECK 'REGISTER.
84	0034	3AC3 FFFF	STAR 3.
85	0035	8000 006E	J CHECK
86	0036	3AC4 FFFF	STAR 4.
87	0037	8000 006E	J CHECK
88	0038	3AC5 FFFF	STAR 5.
89	0039	8000 006E	J CHECK
90	003A	3AC6 FFFF	STAR 6.
91	003B	8000 006E	J CHECK
92	003C	3AC7 FFFF	STAR 7.
93	003D	8000 006E	J CHECK
94	003E	3AC8 FFFF	STAR 8.
95	003F	8000 006E	J CHECK
96	0040	3AC9 FFFF	STAR 9.
97	0041	8000 006E	J CHECK
98	0042	3ACA FFFF	STAR 10.
99	0043	8000 006E	J CHECK
100	0044	3ACB FFFF	STAR 11.
101	0045	8000 006E	J CHECK
102	0046	3ACC FFFF	STAR 12.
103	0047	8000 006E	J CHECK
104	0048	3ACD FFFF	STAR 13.
105	0049	8000 006E	J CHECK
106	004A	3ACE FFFF	STAR 14.

107	004B	8000	006E	J CHECK	
108	004C	3ACF	FFFF	STAR 15.	
109	004D	8000	006E	J CHECK	
110	004E	3AD0	FFFF	STAR 16.	
111	004F	8000	006E	J CHECK	
112	0050	3AD1	FFFF	STAR 17.	
113	0051	8000	006E	J CHECK	
114	0052	3AD2	FFFF	STAR 18.	
115	0053	8000	006E	J CHECK	
116	0054	3AD3	FFFF	STAR 19.	
117	0055	8000	006E	J CHECK	
118	0056	3AD4	FFFF	STAR 20.	
119	0057	8000	006E	J CHECK	
120	0058	3AD5	FFFF	STAR 21.	
121	0059	8000	006E	J CHECK	
122	005A	3AD6	FFFF	STAR 22.	
123	005B	8000	006E	J CHECK	
124	005C	3AD7	FFFF	STAR 23.	
125	005D	8000	006E	J CHECK	
126	005E	3AD8	FFFF	STAR 24.	
127	005F	8000	006E	J CHECK	
128	0060	3AD9	FFFF	STAR 25.	
129	0061	8000	006E	J CHECK	
130	0062	3ADA	FFFF	STAR 26.	
131	0063	8000	006E	J CHECK	
132	0064	3ADB	FFFF	STAR 27.	
133	0065	8000	006E	J CHECK	
134	0066	3ADC	FFFF	STAR 28.	
135	0067	8000	006E	J CHECK	
136	0068	3ADD	FFFF	STAR 29.	
137	0069	8000	006E	J CHECK	
138	006A	3ADE	FFFF	STAR 30.	
139	006B	8000	006E	J CHECK	
140	006C	3ADF	FFFF	STAR 31.	
141	006D	8000	006E	J CHECK	
142	006E	4F43	0000	CHECK\$NOP	'(FOR PROG TEST, HALT)
143					'(TO INSERT ERROR)
144					'(MANUALLY.)
145	006F	7A43	FFFF	LDAL -1	'INITIALIZE REXAM
146	0070	FF00	0000	S REXAM	
147	0071	6A40	0000	CHK1\$ LDAR 0.	'CHECK
148	0072	8000	00EE	J EXAM	'ALL REGISTERS
149	0073	0000	0005	HLT RTEST,0.	'FOR CORRECT
150	0074	6A41	0000	LDAR 1.	'PATTERN.
151	0075	8000	00EE	J EXAM	
152	0076	0001	0005	HLT RTEST,1.	
153	0077	6A42	0000	LDAR 2.	
154	0078	8000	00EE	J EXAM	
155	0079	0002	0005	HLT RTEST,2.	
156	007A	6A43	0000	LDAR 3.	
157	007B	8000	00EE	J EXAM	
158	007C	0003	0005	HLT RTEST,3.	
159	007D	6A44	0000	LDAR 4.	

160	007E	8000	00EE	J EXAM	'TO CHECK REG 4
161	007F	0004	0005	HLT RTEST,4.	'RETURN IF NOT OK
162	0080	6A45	0000	LDAR 5.	'RETURN IF OK
163	0081	8000	00EE	J EXAM	
164	0082	0005	0005	HLT RTEST,5.	
165	0083	6A46	0000	LDAR 6.	
166	0084	8000	00EE	J EXAM	
167	0085	0006	0005	HLT RTEST,6.	
168	0086	6A47	0000	LDAR 7.	
169	0087	8000	00EE	J EXAM	
170	0088	0007	0005	HLT RTEST,7.	
171	0089	6A48	0000	LDAR 8.	
172	008A	8000	00EE	J EXAM	
173	008B	0008	0005	HLT RTEST,8.	
174	008C	6A49	0000	LDAR 9.	
175	008D	8000	00EE	J EXAM	
176	008E	0009	0005	HLT RTEST,9.	
177	008F	6A4A	0000	LDAR 10.	
178	0090	8000	00EE	J EXAM	
179	0091	000A	0005	HLT RTEST,10.	'INSTR
180	0092	6A4B	0000	LDAR 11.	'FOR HALT WITH
181	0093	8000	00EE	J EXAM	'DECIMAL NUMBER
182	0094	000B	0005	HLT RTEST,11.	'IN R FIELD AND
183	0095	6A4C	0000	LDAR 12.	'RTEST IN MA
184	0096	8000	00EE	J EXAM	'FIELD.
185	0097	000C	0005	HLT RTEST,12.	
186	0098	6A4D	0000	LDAR 13.	
187	0099	8000	00EE	J EXAM	
188	009A	000D	0005	HLT RTEST,13.	
189	009B	6A4E	0000	LDAR 14.	
190	009C	8000	00EE	J EXAM	
191	009D	000E	0005	HLT RTEST,14.	
192	009E	6A4F	0000	LDAR 15.	
193	009F	8000	00EE	J EXAM	
194	00A0	000F	0005	HLT RTEST,15.	
195	00A1	6A50	0000	LDAR 16.	
196	00A2	8000	00EE	J EXAM	
197	00A3	0010	0005	HLT RTEST,16.	
198	00A4	6A51	0000	LDAR 17.	
199	00A5	8000	00EE	J EXAM	
200	00A6	0011	0005	HLT RTEST,17.	
201	00A7	6A52	0000	LDAR 18.	
202	00A8	8000	00EE	J EXAM	
203	00A9	0012	0005	HLT RTEST,18.	
204	00AA	6A53	0000	LDAR 19.	
205	00AB	8000	00EE	J EXAM	
206	00AC	0013	0005	HLT RTEST,19.	
207	00AD	6A54	0000	LDAR 20.	
208	00AE	8000	00EE	J EXAM	
209	00AF	0014	0005	HLT RTEST,20.	
210	00B0	6A55	0000	LDAR 21.	
211	00B1	8000	00EE	J EXAM	
212	00B2	0015	0005	HLT RTEST,21.	

213	00B3	6A56	0000	LDAR 22.
214	00B4	8000	00EF	J EXAM
215	00B5	0016	0005	HLT RTEST,22.
216	00B6	6A57	0000	LDAR 23.
217	00B7	8000	00EE	J EXAM
218	00B8	0017	0005	HLT RTEST,23.
219	00B9	6A58	0000	LDAR 24.
220	00BA	8000	00EF	J EXAM
221	00BB	0018	0005	HLT RTEST,24.
222	00BC	6A59	0000	LDAR 25.
223	00BD	8000	00EF	J EXAM
224	00BE	0019	0005	HLT RTEST,25.
225	00BF	6A5A	0000	LDAR 26.
226	00C0	8000	00EF	J EXAM
227	00C1	001A	0005	HLT RTEST,26.
228	00C2	6A5B	0000	LDAR 27.
229	00C3	8000	00EF	J EXAM
230	00C4	001B	0005	HLT RTEST,27.
231	00C5	6A5C	0000	LDAR 28.
232	00C6	8000	00EF	J EXAM
233	00C7	001C	0005	HLT RTEST,28.
234	00C8	6A5D	0000	LDAR 29.
235	00C9	8000	00EF	J EXAM
236	00CA	001D	0005	HLT RTEST,29.
237	00CB	6A5E	0000	LDAR 30.
238	00CC	8000	00EF	J EXAM
239	00CD	001E	0005	HLT RTEST,30.
240	00CE	6A5F	0000	LDAR 31.
241	00CF	8000	00EF	J EXAM
242	00D0	001F	0005	HLT RTEST,31.
243	00D1	CA00	0005	NREGS LD RTEST
244	00D2	4F43	0000	NOP '(FOR PROG TEST HALT)
245	00D3	7903	0001	ADDL 1 'INCREMENT RTEST
246	00D4	FF00	0005	S RTEST
247	00D5	7623	0020	SUBL 32. 'ALL REGISTERS DCNE
248	00D6	20A0	0008	JNE STR4 'NO, REPEAT
249	00D7	CA00	0004	NPATS LD TEST 'YES, GET NEW PATTERN
250	00D8	4F43	0000	NOP '(FOR PROG TEST HALT)
251	00D9	4043	0000	CK 'COMPLEMENT TEST WD
252	00DA	FF00	0004	S TEST
253	00DB	CAC0	0003	LD ALL
254	00DC	4F43	0000	NOP '(FOR PROG TEST HALT)
255	00DD	4043	0000	CK 'COMPLEMENT BACKGND
256	00DE	FF00	0003	S ALL
257	00DF	20A0	0006	JNE STR3 'ALL=0
258	00E0	CA00	0004	LD TEST 'YES
259	00E1	2740	00E5	JBS 17,DSPL 'ALL BITS TESTED
260	00E2	4C03	0000	SL1 'NO, SHIFT TEST BIT 1
261	00E3	FF00	0004	S TEST 'SAVE
262	00E4	8000	0006	J STR3 'REPEAT FOR NEXT BIT
263				'DISPLAY ROUTINE
264	00E5	CA00	0002	DSPLS LD COUNT 'YES, TEST COMPLETE
265	00E6	7903	0001	ADD,L 1 'INCREMENT COUNT

266	00E7	FF00	0002		S COUNT	
267	00E8	3A44	0FFF		LDRL ASTEP,R4	
268	00E9	3A45	0200		LDRL BSTEP,R5	
269	00EA	1144	00EA	PAUS\$	JRNZ,D PAUS,R4	'PAUSE, DISPLAY
270	00EB	3A44	0FFF		LDRL ASTEP,R4	'COUNT IN
271	00EC	1145	00EA		JRNZ,D PAUS,R5	'A REGISTER FOR
272	00ED	8000	0002		J STR2	'THREE SECONDS.
273					.	
274					.	
275					'EXAMINE R CONTENTS ROUTINE	
276	00EE	FF00	0001	EXAMS\$	S R\$AV	'SAVE R VALUE
277	00EF	7A43	0001		LDAL 1	
278	00F0	E900	0000		ADD REXAM	'CALC REG BEING
279	00F1	FF00	0000		S REXAM	'EXAMINED.
280	00F2	4F43	0000		NOP	'(FOR PROG TEST HALT)
281	00F3	7A43	0071		LDAL CHK1	'CALC NORMAL
282	00F4	5905	0000		ADD,2M REXAM	'RETURN ADDRESS.
283	00F5	5903	0000		ADD,M REXAM	
284	00F6	7903	0003		ADD,L 3	
285	00F7	FF00	0006		S WHICH	'SET RETURN ADDRESS
286	00F8	CA00	0005		LD RTEST	'GET RTST VAL
287	00F9	D600	0000		SUB REXAM	'COMPARE REXAM
288	00FA	2080	00FE		JEQ GTST	'REXAM=RTEST
289	00FB	CA00	0003		LD ALL	'NO, GET BACKGND.
290	00FC	00C0	0003		RF 0	'RESET FLAG
291						'TO INDICATE
292						'REGISTER SHOULD
293						'HAVE BACKGND.
294	00FD	8000	0100		J .+3	
295	00FE	CA00	0004	GTSTS	LD TEST	'YES, GET TEST WD.
296	00FF	08C0	0000		SF 0	'SET FLAG
297						'TO INDICATE
298						'REGISTER SHOULD
299						'HAVE TEST WD.
300	0100	D600	0001		SUB R\$AV	'R\$AV CORRECT
301	0101	2080	0106		JEQ RTN	'YES, NORMAL RETURN
302	0102	CA00	0006		LD WHICH	'NO, RETURN TO HALT
303	0103	7623	0001		SUB,L 1	
304	0104	FF00	0006		S WHICH	
305	0105	CA00	0004		LD TEST	'TEST WD TO
306						'A REGISTER
307						'FOR DISPLAY
308	0106	8080	0006	RTNS J	WHICH	'RETURN
309					.	
310					.	
311		00000000			.DATA 0	
312	0000	0000	0000	REXAMS	0	
313	0001	0000	0000	R\$AVS	0	
314	0002	0000	0000	COUNTS	0	
315	0003	0000	0000	ALLS	0	
316	0004	0000	0000	TESTS	0	
317	0005	0000	0000	RTESTS	0	
318	0006	0000	0000	WHICHS	0	

# INSTRUCTION TEST PROGRAM - HFDR

```

10      ..... ! PUSH PM CLEAR
20      ..... ! LOAD PROGRAM
30      00000000 .PROG 0
40      0000 8000 0504 JMP WXYZ
50      0001 8000 050A JMP ZYXW
60      0002 0000 0000 HLT
70      0003 0000 0000 HLT
80      0004      STRTS
90      0004 08E0 0000 DINT
100     0005 FF00 0000 STA RM1
110     0006 FF00 0001 STA RM2
120     0007 FF00 0003 STA EMTY
130     0008 3AC1 FFFF STAR 1
140     0009 3AC2 FFFF STAR 2
150     000A 3AC3 FFFF STAR 3
160     000B CA00 0002 LDA FULL
170     000C FF00 0003 STA EMTY
180     000D 3AC4 FFFF STAR 4
190     000E 2080 053C JEQ OUT2
200     000F 0000 0000 HLT !HLT WORKED A=EMTY=R4=FFFF R1=R2=R3=RM1=RM2=0000
210
220     0010      GO1$
230     0010 1404 0538 JRND OUT1,4
240     0011 1403 0013 JRND .+2,R3
250     0012 0000 0000 HLT
260     0013 7903 0001 ADDL 1
270     0014 2080 0016 JEQ ONE
280     0015 0000 0000 HLT
290     0016      ONE$
300     0016 CA00 0000 LDA RM1
310     0017 7903 0001 ADDL 1
320     0018 FF00 0000 STA RM1
330     0019 CA00 0001 LDA RM2
340     001A 7623 0001 SUBL 1
350     001B FF00 0001 STA RM2
360     001C E900 0000 ADD RM1
370     001D 2080 0022 JEQ TWO
380     001E 7903 8000 ADDL 100000
390     001F 7903 8000 ADDL 100000
400     0020 2080 0022 JEQ .+2
410     0021 0000 0000 HLT
420     0022      TWO$
430     0022 3901 0001 XR,ADD,L 1,R1
440     0023 6A41 0000 LDAR R1
450     0024 3AC2 FFFF STAR R2
460     0025 36A2 FFFF XR,SUB,A R2
470     0026 6A42 0000 LDAR R2
480     0027 2080 0029 JEQ .+2
490     0028 0000 0000 HLT
500     0029 1401 0016 JRND ONE,R1
510
520     002A 7A43 0000 LDAL 0
530     002B 7903 0000 ADDL 0

```

540	002C	2080	002E	JEQ .+2
550	002D	0000	0000	HLT
560				' F4 F5 F6 F7
570	002E	4982	0000	ADD,CA,B
580	002F	7903	0001	ADDL 1
590	0030	2080	0032	JEQ .+2
600	0031	0000	0000	HLT
610	0032	7A43	FFFF	LDAL 177777
620	0033	4982	0000	ADD,CA,B
630	0034	7903	0001	ADDL 1
640	0035	2080	0037	JEQ .+2
650	0036	0000	0000	HLT
660	0037	7A43	AAAA	LDAL AAAA
670	0038	4984	0000	ADD,CA,2B
680	0039	7903	0001	ADDL 1
690	003A	7623	AAAA	SUBL AAAA
700	003B	2080	003D	JEQ .+2
710	003C	0000	0000	HLT
720	003D	7A43	5555	LDAL 52525
730	003E	4984	0000	ADD,CA,2B
740	003F	7903	0001	ADDL 1
750	0040	7623	5555	SUBL 52525
760	0041	2080	0043	JEQ .+2
770	0042	0000	0000	HLT
780	0043	5903	0002	ADD,A,M FULL
790	0044	7903	0001	ADDL 1
800	0045	2080	0047	JEQ .+2
810	0046	0000	0000	HLT
820	0047	7A43	FFFF	LDAL FFFF
830	0048	5903	0007	ADD,A,M ZERO
840	0049	7903	0001	ADDL 1
850	004A	2080	004C	JEQ .+2
860	004B	0000	0000	HLT
870	004C	7A43	AAAA	LDAL AAAA
880	004D	5984	0006	ADD,CA,2M MEMO
890	004E	7623	AAAA	SUBL AAAA
900	004F	2080	0051	JEQ .+2
910	0050	0000	0000	HLT
920	0051	7A43	5555	LDAL 52525
930	0052	5984	0008	ADD,CA,2M OMEM
940	0053	7623	5554	SUBL 52524
950	0054	2080	0056	JEQ .+2
960	0055	0000	0000	HLT
970	0056	7A43	FFFF	LDAL FFFF
980	0057	3AC1	FFFF	STAR R1
990	0058	7A43	0001	LDAL 1
1000	0059	6901	0000	ADD,A,R R1
1010	005A	2080	005C	JEQ .+2
1020	005B	0000	0000	HLT
1030	005C	3AC1	FFFF	STAR R1
1040	005D	6901	0000	ADD,A,R R1
1050	005E	2080	0060	JEQ .+2
1060	005F	0000	0000	HLT

1070	0060	7905	1999	ADD,A,2L 14631
1080	0061	7623	3332	SUBL 31462
1090	0062	2080	0064	JEQ .+2
1100	0063	0000	0000	HLT
1110	0064	7905	7777	ADD,A,2L 73567
1120	0065	7623	EEEE	SUBL 167356
1130	0066	7903	8000	ADDL 100000
1140	0067	7903	8000	ADDL 100000
1150	0068	2080	006A	JEQ .+2
1160	0069	0000	0000	HLT
1170	006A	7905	5555	ADD,A,2L 52525
1180	006B	7623	AAAA	SUBL AAAA
1190	006C	7903	8000	ADDL 100000
1200	006D	7903	8000	ADDL 100000
1210	006E	2080	0070	JEQ .+2
1220	006F	0000	0000	HLT
1230				* -1
1240	0070	7A43	0001	LDAL 1
1250				* ADD,A,R=-1
1260	0071	4901	0000	.INSTRT 44401
1270	0072	2080	0074	JEQ .+2
1280	0073	0000	0000	HLT
1290				* 28B
1300	0074	7A43	FFFF	LDAL FFFF
1310				* ADD,ZK,28B
1320	0075	4987	0000	.INSTRT 44607
1330	0076	7903	0002	ADDL 2
1340	0077	2080	0079	JEQ .+2
1350	0078	0000	0000	HLT
1360	0079	7A43	AAAA	LDAL 125252
1370				* ADD,ZK,28B
1380	007A	4987	0000	.INSTRT 44607
1390	007B	7903	8000	ADDL 100000
1400	007C	7903	8000	ADDL 100000
1410	007D	2080	007F	JEQ .+2
1420	007E	0000	0000	HLT
1430	007F	7A43	0000	LDAL 0
1440				* ADD,ZK,28B
1450	0080	4987	0000	.INSTRT 44607
1460	0081	2080	0083	JEQ .+2
1470	0082	0000	0000	HLT
1480				* B=A
1490	0083	7A43	FFFF	LDAL FFFF
1500				* ADD,OK,B=A
1510	0084	4910	0000	.INSTRT 44420
1520	0085	7903	0002	ADDL 2
1530	0086	2080	0088	JEQ .+2
1540	0087	0000	0000	HLT
1550				* ADD,CA,B=A
1560	0088	4990	0000	.INSTRT 44620
1570	0089	7903	0001	ADDL 1
1580	008A	2080	003C	JEQ .+2
1590	008B	0000	0000	HLT

1600	008C	7A43	0C00	LDAL 0
1610				' ADD,CA,B=AB
1620	008D	4592	0000	.INSTRT 44622
1630	008E	7903	0001	ADDL 1
1640	008F	2080	0091	JEQ .+2
1650	0090	0000	0000	HLT
1660				' ADD,ZK,B=A
1670	0091	4991	0000	.INSTRT 44621
1680	0092	2080	0094	JEQ .+2
1690	0093	0000	0000	HLT
1700	0094	7A43	FFFF	LDAL FFFF
1710	0095	7903	0000	ADDL 0
1720				' ADD,ZK,B=AB
1730	0096	4592	0000	.INSTRT 44622
1740	0097	2080	0099	JEQ .+2
1750	0098	0000	0000	HLT
1760	0099	7A43	0001	LDAL 1
1770	009A	7903	FFFF	ADDL FFFF
1780				' ADD,A,B=AB
1790	009B	4913	0000	.INSTRT 44423
1800	009C	2080	009E	JEQ .+2
1810	009D	0000	0000	HLT
1820	009E	7A43	FFFF	LDAL FFFF
1830				' ADD,A,AB
1840	009F	4913	0000	.INSTRT 44423
1850	00A0	7903	0002	ADDL 2
1860	00A1	2080	00A3	JEQ .+2
1870	00A2	0000	0000	HLT
1880	00A3	7A43	FFFF	LDAL FFFF
1890				' ADD,A,B,=A2B
1900	00A4	4915	0000	.INSTRT 44425
1910	00A5	7903	0003	ADDL 3
1920	00A6	2080	00A8	JEQ .+2
1930	00A7	0000	0000	HLT
1940	00A8	7A43	5555	LDAL 52525
1950				' ADD,ZK,B=2AB
1960	00A9	4995	0000	.INSTRT 44625
1970	00AA	2080	00AC	JEQ .+2
1980	00AB	0000	0000	HLT
1990	00AC	7A43	AAAA	LDAL AAAA
2000				' ADD,ZK,B=2AB
2010	00AD	4995	0000	.INSTRT 44625
2020	00AE	7903	8000	ADDL 100000
2030	00AF	7903	8000	ADDL 100000
2040	00B0	2080	00B2	JEQ .+2
2050	00B1	0000	0000	HLT
2060	00B2	7A43	FFFF	LDAL FFFF
2070				' ADD,ZK,B=A2BB
2080	00B3	4997	0000	.INSTRT 44627
2090	00B4	7903	0002	ADDL 2
2100	00B5	2080	00B7	JEQ .+2
2110	00B6	0000	0000	HLT
2120	00B7	7A43	5555	LDAL 52525

2130				* ADD,ZK,B=A288
2140	0088	4997	0000	.INSTRT 44627
2150	0089	2080	0088	JEQ .+2
2160	008A	0000	0000	HLT
2170	008B	7A43	AAAA	LDAL AAAA
2180				* ADD,ZK,B=A28H
2190	008C	4997	0000	.INSTRT 44627
2200	008D	7903	8000	ADDL 100000
2210	008E	7903	8000	ADDL 100000
2220	008F	2080	00C1	JEQ .+2
2230	00C0	0000	0000	HLT
2240				* F5
2250				* ADD,OK,B=OK
2260	00C1	5900	0000	.INSTRT 5440C
2270	00C2	7903	0002	ADDL 2
2280	00C3	2080	00C5	JEQ .+2
2290	00C4	0000	0000	HLT
2300				* ADD,ZK,B=OK
2310	00C5	5981	0000	.INSTR 54601
2320	00C6	7903	0001	ADDL 1
2330	00C7	2080	00C9	JEQ .+2
2340	00C8	0000	0000	HLT
2350				* ADD,ZK,B=2DMDM DM=FFFF
2360	00C9	5987	0004	.INSTRT 54607,FRST
2370	00CA	7903	0002	ADDL 2
2380	00CB	2080	00CD	JEQ .+2
2390	00CC	0000	0000	HLT
2400				* ADD,ZK,B=2DMDM
2410	00CD	5987	0005	.INSTRT 54607,SND
2420	00CE	2080	0000	JEQ .+2
2430	00CF	0000	0000	HLT
2440				* ADD,ZK,B=2DMDM DM=AAAA
2450	00D0	5987	0000	.INSTRT 54607,TRE
2460	00D1	7903	8000	ADDL 100000
2470	00D2	7903	8000	ADDL 100000
2480	00D3	2080	00D5	JEQ .+2
2490	00D4	0000	0000	HLT
2500				* F6 DONE
2510				* F7
2520				* ADD,ZK,B=OK
2530	00D5	7981	0000	.INSTRT 74601
2540	00D6	7903	0001	ADDL 1
2550	00D7	2080	00D9	JEQ .+2
2560	00D8	0000	0000	HLT
2570				* ADD,OK,B=OK
2580	00D9	7900	0000	.INSTRT 74400
2590	00DA	7903	0002	ADDL 2
2600	00DB	2080	0000	JEQ .+2
2610	00DC	0000	0000	HLT
2620				* ADD,ZK,B=2LL L=FFFF
2630	00DD	7987	FFFF	.INSTRT 74607,FFFF
2640	00DE	7903	0002	ADDL 2
2650	00DF	2080	00E1	JEQ .+2

2660	00E0	0000	0000	HLT
2670				* ADD,ZK,8=2LL 6=5555
2680	00E1	7987	5555	.INSTRT 74607,52525
2690	00E2	2080	00E4	JEQ .+2
2700	00E3	0000	0000	HLT
2710	00E4	7987	AAAA	.INSTRT 74607,AAAA
2720	00E5	7903	8000	ADDL 100000
2730	00E6	7903	8000	ADDL 100000
2740	00E7	2080	00E9	JEQ .+2
2750	00E8	0000	0000	HLT
2760				* PARITY CHECK
2770	00E9	7A43	0000	LDAL 0
2780				* ADD,CA,8,FORM PARITY
2790				* ADD,CA,8,FORM PARITY
2800				* ADD,CA,8,FORM PARITY
2810	00EA	4963	0000	.INSTRT 44543
2820				* JMP ODD PARITY OUTA
2830	00EB	2580	0540	.INSTRT 22600,OUTA
2840				* JMP EVEN PARITY .+2
2850	00EC	25A0	00EF	.INSTRT 22640,..+2
2860	00ED	0000	0000	HLT
2870	00EE	7A43	FFFF	LDAL FFFF
2880				* ADD,CA,8,FORM PARITY
2890	00EF	4963	0000	.INSTRT 44543
2900				* JMP ODD PARITY OUTB
2910	00F0	2580	0541	.INSTRT 22600,OUTB
2920				* JMP EVEN PARITY .+2
2930	00F1	25A0	00F3	.INSTRT 22640,..+2
2940	00F2	0000	0000	HLT
2950	00F3	7A43	AAAA	LDAL AAAA
2960				* ADD,CA,8,FORM PARITY
2970	00F4	4963	0000	.INSTRT 44543
2980				* JMP ODD PARITY OUTC
2990	00F5	2580	0542	.INSTRT 22600,OUTC
3000				* JMP EVEN PARITY .+2
3010	00F6	25A0	00F8	.INSTRT 22640,..+2
3020	00F7	0000	0000	HLT
3030	00F8	7A43	5555	LDAL 52525
3040				* ADD,CA,8,FORM PARITY
3050	00F9	4963	0000	.INSTRT 44543
3060				* JMP ODD PARITY OUTD
3070	00FA	2580	0543	.INSTRT 22600,OUTD
3080				* JMP EVEN PARITY .+2
3090	00FB	25A0	00FD	.INSTRT 22640,..+2
3100	00FC	0000	0000	HLT
3110	00FD	7A43	8000	LDAL 100000
3120				* ADD,CA,8,FORM PARITY
3130	00FE	4963	0000	.INSTRT 44543
3140	00FE	25A0	0544	.INSTRT 22640,OUTE
3150				* JMP ODD PARITY .+2
3160	0100	2580	0102	.INSTRT 22600,..+2
3170	0101	0000	0000	HLT
3180	0102	7A43	2000	LDAL 20000

3190				* ADD,CA,B,FORM PARITY
3200	0103	4963	0000	.INSTRT 44543
3210	0104	25A0	0545	.INSTRT 22640,OUTF
3220				* JMP ODD PARITY .+2
3230	0105	2580	C107	.INSTRT 22600,..+2
3240	0106	0000	0000	HLT
3250	0107	7A43	0800	LDAL 4000
3260				* ADD,CA,B,FORM PARITY
3270	0108	4963	0000	.INSTRT 44543
3280	0109	25A0	0546	.INSTRT 22640,OUTG
3290				* JMP ODD PARITY .+2
3300	010A	2580	C10C	.INSTRT 22600,..+2
3310	010B	0000	0000	HLT
3320	010C	7A43	0200	LDAL 1000
3330				* ADD,CA,B,FORM PARITY
3340	010D	4963	0000	.INSTRT 44543
3350	010E	25A0	0547	.INSTRT 22640,OUTH
3360				* JMP ODD PARITY .+2
3370	010F	2580	0111	.INSTRT 22600,..+2
3380	0110	0000	0000	HLT
3390	0111	7A43	0080	LDAL 200
3400				* ADD,CA,B,FORM PARITY
3410	0112	4963	0000	.INSTRT 44543
3420	0113	25A0	0548	.INSTRT 22640,OUTI
3430				* JMP ODD PARITY .+2
3440	0114	2580	0116	.INSTRT 22600,..+2
3450	0115	0000	0000	HLT
3460	0116	7A43	0020	LDAL 40
3470				* ADD,CA,B,FORM PARITY
3480	0117	4963	0000	.INSTRT 44543
3490	0118	25A0	C549	.INSTRT 22640,OUTJ
3500				* JMP ODD PARITY .+2
3510	0119	2580	C118	.INSTRT 22600,..+2
3520	011A	0000	0000	HLT
3530	011B	7A43	C008	LDAL 10
3540				* ADD,CA,B,FORM PARITY
3550	011C	4963	0000	.INSTRT 44543
3560	011D	25A0	054A	.INSTRT 22640,OUTK
3570				* JMP ODD PARITY .+2
3580	011E	2580	C120	.INSTRT 22600,..+2
3590	011F	0000	0000	HLT
3600	0120	7A43	0002	LDAL 2
3610				* ADD,CA,B,FORM PARITY
3620	0121	4963	0000	.INSTRT 44543
3630	0122	25A0	054B	.INSTRT 22640,OUTL
3640				* JMP ODD PARITY .+2
3650	0123	2580	C125	.INSTRT 22600,..+2
3660	0124	0000	0000	HLT
3670	0125	7A43	0C00	LDAL 0
3680	0126	490B	0000	ADD,A,B,SR2
3690	0127	2C80	0129	JEO .+2
3700	0128	0000	0000	HLT
3710	0129	7A43	FFFF	LDAL FFFF

3720	012A	498B	0000	ADD,ZK,B,SR2
3730	012B	7903	0001	ADDL 1
3740	012C	2080	012E	JEQ .+2
3750	012D	0000	0000	HLT
3760	012E	7A43	5555	LDAL 52525
3770	012F	498B	0000	ADD,ZK,B,SR2
3780	0130	7623	1555	SUBL 12525
3790	0131	2080	0133	JEQ .+2
3800	0132	0000	0000	HLT
3810	0133	7A43	7FFF	LDAL 77777
3820	0134	498B	0000	ADD,ZK,B,SR2
3830	0135	7623	1FFF	SUBL 17777
3840	0136	2080	0138	JEQ .+2
3850	0137	0000	0000	HLT
3860	0138	7A43	6666	LDAL 63146
3870	0139	498B	0000	ADD,ZK,B,SR2
3880	013A	7623	1999	SUBL 14631
3890	013B	2080	013D	JEQ .+2
3900	013C	0000	0000	HLT
3910	013D	7A43	1999	LDAL 14631
3920	013E	498B	0000	ADD,ZK,B,SR2
3930	013F	7623	0666	SUBL 3146
3940	0140	2080	0142	JEQ .+2
3950	0141	0000	0000	HLT
3960				' FLAG TEST
3970	0142	08C0	0000	SF 0
3980	0143	09C0	0000	SF 1
3990	0144	0AC0	0000	SF 2
4000	0145	0BC0	0000	SF 3
4010	0146	0CC0	0000	SF 4
4020	0147	0DC0	0000	SF 5
4030	0148	0ECC	0000	SF 6
4040	0149	0FC0	0000	SF 7
4050	014A	7A43	FFFF	LDAL FFFF
4060	014B	3AC1	FFFF	STAR 1
4070	014C			FLAGS
4080	014C	0000	0000	RF 0
4090	014D	20C0	052B	JFS 0,AOUT
4100	014E	20E0	0150	JFR 0,..+2
4110	014F	0000	0000	HLT
4120	0150	01C0	0000	RF 1
4130	0151	21C0	052C	JFS 1,AOUT
4140	0152	21E0	0154	JFR 1,..+2
4150	0153	0000	0000	HLT
4160	0154	02C0	0000	RF 2
4170	0155	22C0	052D	JFS 2,COUT
4180	0156	22E0	0158	JFR 2,..+2
4190	0157	0000	0000	HLT
4200	0158	03C0	0000	RF 3
4210	0159	23C0	052E	JFS 3,DOUT
4220	015A	23E0	015C	JFR 3,..+2
4230	015B	0000	0000	HLT
4240	015C	04C0	0000	RF 4

4250	015D	24C0	052F	JFS 4,EOUT
4260	015E	24E0	0160	JFR 4,..+2
4270	015F	0000	0000	HLT
4280	0160	05C0	0000	RF 5
4290	0161	25C0	0530	JFS 5,FOUT
4300	0162	25E0	0164	JFR 5,..+2
4310	0163	0000	0000	HLT
4320	0164	06C0	0000	RF 6
4330	0165	26C0	0531	JFS 6,GOUT
4340	0166	26E0	0168	JFR 6,..+2
4350	0167	0000	0000	HLT
4360	0168	07C0	0000	RF 7
4370	0169	27C0	0532	JFS 7,HOUT
4380	016A	27E0	016C	JFR 7,..+2
4390	016B	0000	0000	HLT
4400	016C	6A41	0000	LDAR 1
4410	016D	2080	0173	JEQ .+6
4420	016E	7903	0001	ADDL 1
4430	016F	3AC1	FFFF	STAR 1
4440	0170	6A41	0000	LDAR 1
4450	0171	2080	014C	JEQ FLAG
4460	0172	0000	0000	HLT
4470	0173	7A43	FFFF	LDAL FFFF
4480	0174	3AC1	FFFF	STAR 1
4490	0175			GLAGS
4500	0175	08C0	0000	SF 0
4510	0176	20E0	0533	JFR 0,IOUT
4520	0177	20C0	0179	JFS 0,..+2
4530	0178	0000	0000	HLT
4540	0179	09C0	0000	SF 1
4550	017A	21E0	0534	JFR 1,JOUT
4560	017B	21C0	017D	JFS 1,..+2
4570	017C	0000	0000	HLT
4580	017D	0AC0	0000	SF 2
4590	017E	22E0	0535	JFR 2,KOUT
4600	017F	22C0	0181	JFS 2,..+2
4610	0180	0000	0000	HLT
4620	0181	08C0	0000	SF 3
4630	0182	23E0	0536	JFR 3,LOUT
4640	0183	23C0	0185	JFS 3,..+2
4650	0184	0000	0000	HLT
4660	0185	00C0	0000	SF 4
4670	0186	24E0	0537	JFR 4,MOUT
4680	0187	24C0	0189	JFS 4,..+2
4690	0188	0000	0000	HLT
4700	0189	0DC0	0000	SF 5
4710	018A	25E0	0538	JFR 5,NOUT
4720	018B	25C0	018D	JFS 5,..+2
4730	018C	0000	0000	HLT
4740	018D	0EC0	0000	SF 6
4750	018E	26E0	0539	JFR 6,OOUT
4760	018F	26C0	0191	JFS 6,..+2
4770	0190	0000	0000	HLT

4780	0191	0FC0	0000	SF 7
4790	0192	27E0	053A	JFR 7,POUT
4800	0193	27C0	0195	JFS 7,+.2
4810	0194	0000	0000	HLT
4820	0195	6A41	0000	LDAR 1
4830	0196	2080	019C	JEQ .+6
4840	0197	79C3	0C01	ADDL 1
4850	0198	3AC1	FFFF	STAR 1
4860	0199	6A41	0000	LDAR 1
4870	019A	2080	0175	JEQ GLAG
4880	019B	0000	0000	HLT
4890				* F1
4900	019C	7A43	FFFF	LUAL FFFE
4910	019D	3AC1	FFFF	STAR 1
4920	019E	3AC3	FFFF	STAR 3
4930	019F	7A43	0C02	LDAL 2
4940	01A0	3AC2	FFFF	STAR 2
4950	01A1	3AC4	FFFF	STAR 4
4960	01A2	3AC5	FFFF	STAR 5
4970	01A3			BEGs
4980	01A3	1141	01A5	JRNZ,D .+2,1
4990	01A4	0000	0000	HLT
5000	01A5	1162	01A7	JRNZ,I .+2,2
5010	01A6	0000	0000	HLT
5020	01A7	1443	01A9	JRNO,D .+2,3
5030	01A8	0000	0000	HLT
5040	01A9	1464	01AB	JRNO,I .+2,4
5050	01AA	0000	0000	HLT
5060	01AB	18C1	0552	JRO OA,1
5070	01AC	1202	0553	JRZ OB,2
5080	01AD	1102	01B1	JRNZ .+4,2
5090	01AE	6A42	0000	LDAR 2
5100	01AF	2080	01B1	JEQ .+2
5110	01B0	0000	0000	HLT
5120	01B1	6A45	0000	LDAR 5
5130	01B2	7903	0001	ADDL 1
5140	01B3	3AC5	FFFF	STAR 5
5150	01B4	6901	0000	ADDR 1
5160	01B5	2080	01B7	JEQ .+2
5170	01B6	0000	0000	HLT
5180	01B7	6A42	0000	LDAR 2
5190	01B8	6625	0000	SUBR 5
5200	01B9	2080	01B8	JEQ .+2
5210	01BA	0000	0000	HLT
5220	01BB	6A43	0000	LDAR 3
5230	01BC	6905	0000	ADDR 5
5240	01BD	2080	01C2	JEQ .+5
5250	01BE	7903	8000	ADDL 100000
5260	01BF	7903	8000	ADDL 100000
5270	01C0	2080	01C2	JEQ .+2
5280	01C1	0000	0000	HLT
5290	01C2	6A44	0000	LDAR 4
5300	01C3	6625	0000	SUBR 5

5310	01C4	2080	01C6	JEQ .+2
5320	01C5	0000	0000	HLT
5330	01C6	1402	01A3	JRNO BEG,2
5340	01C7	1802	01C9	JRO .+2,2
5350	01C8	0000	0000	HLT
5360	01C9	1141	01C8	JRNZ,0 .+2,1
5370	01CA	0000	0000	HLT
5380	01CB	1201	01CD	JRZ .+2,1
5390	01CC	0000	0000	HLT
5400	01CD	1101	0554	JRNZ OC,1
5410	01CE	1141	0555	JRNZ,D OD,1
5420	01CF	1161	0556	JRNZ,I OE,1
5430	01D0	1442	0557	JRNC,D OF,2
5440	01D1	1462	0558	JRNO,I OG,2
5450	01D2	1142	01D4	JRNZ,D .+2,2
5460	01D3	0000	0000	HLT
5470	01D4	3902	0002	XR,ADD,L 2,2
5480	01D5	1462	01D7	JRNO,I .+2,2
5490	01D6	0000	0000	HLT
5500	01D7	1462	01D9	JRNO,I .+2,2
5510	01D8	0000	0000	HLT
5520	01D9	3622	0002	XR,SUB,L 2,2
5530	01DA	1202	01DC	JRZ .+2,2
5540	01DB	0000	0000	HLT
5550				' F2
5560	01DC	7A43	0001	LDAL 1
5570	01DD	2020	0559	JBR 0,AX
5580	01DE	2000	01E0	JBS 0,..+2
5590	01DF	0000	0000	HLT
5600	01E0	7A43	0002	LDAL 2
5610	01E1	2100	01E3	JBS 1,..+2
5620	01E2	0000	0000	HLT
5630	01E3	7A43	0004	LDAL 4
5640	01E4	2200	01E6	JBS 2,..+2
5650	01E5	0000	0000	HLT
5660	01E6	7A43	0008	LDAL 8.
5670	01E7	2300	01E9	JBS 3,..+2
5680	01E8	0000	0000	HLT
5690	01E9	7A43	0010	LDAL 16.
5700	01EA	2400	01EC	JBS 4,..+2
5710	01EB	0000	0000	HLT
5720	01EC	7A43	0020	LDAL 32.
5730	01ED	2500	01EF	JBS 5,..+2
5740	01EE	0000	0000	HLT
5750	01EF	7A43	0040	LDAL 64.
5760	01F0	2600	01F2	JBS 6,..+2
5770	01F1	0000	0000	HLT
5780	01F2	7A43	0080	LDAL 128.
5790	01F3	2700	01F5	JBS 7,..+2
5800	01F4	0000	0000	HLT
5810	01F5	7A43	0100	LDAL 256.
5820	01F6	2060	055A	JBR 8.,BX
5830	01F7	2040	01F9	JBS 8,..+2

5840	01F8	0000	0000	HLT
5850	01F9	7A43	0200	LDAL 512.
5860	01FA	2140	01FC	JBS 9...+2
5870	01FB	0000	0000	HLT
5880	01FC	7A43	0400	LDAL 1024.
5890	01FD	2240	01FF	JBS 10...+2
5900	01FE	0000	0000	HLT
5910	01FF	7A43	0800	LDAL 2048.
5920	0200	2340	0202	JBS 11...+2
5930	0201	0000	0000	HLT
5940	0202	7A43	1000	LDAL 4096.
5950	0203	2440	0205	JBS 12...+2
5960	0204	0000	0000	HLT
5970	0205	7A43	2000	LDAL 8192.
5980	0206	2540	0208	JBS 13...+2
5990	0207	0000	0000	HLT
6000	0208	7A43	4000	LDAL 16384.
6010	0209	2640	0208	JBS 14...+2
6020	020A	0000	0000	HLT
6030	020B	7A43	8000	LDAL 32768.
6040	020C	2740	020E	JBS 15...+2
6050	020D	0000	0000	HLT
6060	020E	7A43	FFFF	LDAL -2
6070	020F	2000	0558	JBS 0,CX
6080	0210	2020	0212	JBR 0...+2
6090	0211	0000	0000	HLT
6100	0212	7A43	FFFF	LDAL -3
6110	0213	2120	0215	JBR 1...+2
6120	0214	0000	0000	HLT
6130	0215	7A43	FFF8	LDAL -5
6140	0216	2220	0218	JBR 2...+2
6150	0217	0000	0000	HLT
6160	0218	7A43	FFF7	LDAL -9.
6170	0219	2320	021B	JBR 3...+2
6180	021A	0000	0000	HLT
6190	021B	7A43	FFEF	LDAL -17.
6200	021C	2420	021E	JBR 4...+2
6210	021D	0000	0000	HLT
6220	021E	7A43	FFDF	LDAL -33.
6230	021F	2520	0221	JBR 5...+2
6240	0220	0000	0000	HLT
6250	0221	7A43	FFBF	LDAL -65.
6260	0222	2620	0224	JBR 6...+2
6270	0223	0000	0000	HLT
6280	0224	7A43	FF7F	LDAL -129.
6290	0225	2720	0227	JBR 7...+2
6300	0226	0000	0000	HLT
6310	0227	7A43	FEFF	LDAL -257.
6320	0228	2040	055C	JBS 8...DX
6330	0229	2060	022B	JBR 8...+2
6340	022A	0000	0000	HLT
6350	022B	7A43	FDFE	LDAL -513.
6360	022C	2160	022E	JBR 9...+2

6370	022D	000C	0000	HLT
6380	022E	7A43	FRFF	LDAL -1075.
6390	022F	2260	0231	JBR 10...+2
6400	0230	000C	0000	HLT
6410	0231	7A43	F7FF	LDAL -2049.
6420	0232	2360	0234	JBR 11...+2
6430	0233	000C	0000	HLT
6440	0234	7A43	FFFF	LDAL -4097.
6450	0235	2460	0237	JBR 12...+2
6460	0236	000C	0000	HLT
6470	0237	7A43	DFFF	LDAL -8193.
6480	0238	2560	023A	JBR 13...+2
6490	0239	000C	0000	HLT
6500	023A	7A43	BFFF	LDAL -16385.
6510	023B	2660	023D	JBR 14...+2
6520	023C	0000	0000	HLT
6530	023D	7A43	7FFF	LDAL -32769.
6540	023F	2760	0240	JBR 15...+2
6550	023F	000C	0000	HLT
6560	0240	7A43	0000	LDAL 0
6570				* ADD,CA,B,FCRM PARITY
6580	0241	4963	0000	.INSTRT 44543
6590				* JMP EVEN PAR
6600	0242	25A0	0244	.INSTRT 22640...+2
6610	0243	000C	0000	HLT
6620	0244	7A43	0001	LDAL 1
6630	0245	2080	055D	JEQ ZA
6640	0246	20A0	0248	JNE .+2
6650	0247	000C	0000	HLT
6660	0248	7A43	0002	LDAL 2
6670	0249	2080	055E	JEQ ZB
6680	024A	7A43	0004	LDAL 4
6690	024B	2080	055F	JEQ ZC
6700	024C	7A43	0008	LDAL 8.
6710	024D	2080	0560	JEQ ZD
6720	024E	7A43	0010	LDAL 16.
6730	024F	2080	0561	JEQ ZE
6740	0250	7A43	0020	LDAL 32.
6750	0251	2080	0562	JEQ ZF
6760	0252	7A43	0040	LDAL 64.
6770	0253	2080	0563	JEQ ZG
6780	0254	7A43	0080	LDAL 128.
6790	0255	2080	0564	JEQ ZH
6800	0256	7A43	0100	LDAL 256.
6810	0257	2080	0565	JEQ ZI
6820	0258	7A43	0200	LDAL 512.
6830	0259	2080	0566	JEQ ZJ
6840	025A	7A43	0400	LDAL 1024.
6850	025B	2080	0567	JEQ ZK
6860	025C	7A43	0800	LDAL 2048.
6870	025D	2080	0568	JEQ ZL
6880	025E	7A43	1000	LDAL 4096.
6890	025F	2080	0569	JEQ ZM

6900	0260	7A43	2000	LDAL 8192.
6910	0261	2080	056A	JEQ ZN
6920	0262	7A43	4000	LDAL 16384.
6930	0263	2080	056B	JEQ ZO
6940	0264	7A43	8000	LDAL 32768.
6950	0265	2080	056C	JEQ ZP
6960	0266	7A43	0000	LDAL 0
6970	0267	2380	056D	JOF ZQ
6980	0268	23A0	026A	JNOF .+2
6990	0269	0000	0000	HLT
7000	026A	7A43	FFFF	LDAL FFFF
7010	026B	2380	056E	JOF ZR
7020	026C	23AC	026E	JNOF .+2
7030	026D	0000	0000	HLT
7040	026E	7A43	7FFF	LDAL 77777
7050	026F	7903	0001	ADDL 1
7060	0270	2380	0272	JOF .+2
7070	0271	0000	0000	HLT
7080	0272	23A0	0571	JNOF ZU
7090	0273	7903	7FFF	ADDL 77777
7100	0274	7903	0001	ADDL 1
7110	0275	2380	0277	JOF .+2
7120	0276	0000	0000	HLT
7130	0277	23A0	0572	JNOF ZV
7140	0278	7A43	0000	LDAL 0
7150	0279	2280	027B	JGE .+2
7160	027A	0000	0000	HLT
7170	027B	2180	027D	JLE .+2
7180	027C	0000	0000	HLT
7190	027D	21A0	0575	JGT ZY
7200	027E	22AC	0576	JLT ZZ
7210	027F	7A43	7FFF	LDAL 77777
7220	0280	2280	0282	JGE .+2
7230	0281	0000	0000	HLT
7240	0282	2180	0573	JLE ZW
7250	0283	21A0	0285	JGT .+2
7260	0284	0000	0000	HLT
7270	0285	22AC	0577	JLT ZAA
7280	0286	7A43	FFFF	LDAL FFFF
7290	0287	2180	0289	JLF .+2
7300	0288	0000	0000	HLT
7310	0289	2280	0574	JGE ZX
7320	028A	22AC	028C	JLT .+2
7330	028B	0000	0000	HLT
7340	028C	21A0	0578	JGT ZAB
7350				' F3
7360	028D	3A41	0000	LDRL 0,1
7370	028E	6A41	0000	LDAR 1
7380	028F	2080	0291	JEQ .+2
7390	0290	0000	0000	HLT
7400				' L
7410	0291	3641	FFFF	XR,EOR,L FFFF,1
7420	0292	1801	0294	JRD .+2,1

7430	0293	0000	0000	HLT
7440	0294	3641	FFFF	XR,EOR,L FFFF,1
7450	0295	1201	0297	JRZ .+2,1
7460	0296	0000	0000	HLT
7470	0297	3641	0000	XR,EOR,L 0,1
7480	0298	1201	029A	JRZ .+2,1
7490	0299	0000	0000	HLT
7500				* AL
7510	029A	7A43	0000	LDAL 0
7520	029B	36C1	0000	XR,EOR,AL 0,1
7530	029C	1201	029F	JRZ .+2,1
7540	029D	0000	0000	HLT
7550	029E	36C1	FFFF	XR,EOR,AL FFFF,1
7560	029F	1201	02A1	JRZ .+2,1
7570	02A0	0000	0000	HLT
7580	02A1	7A43	FFFF	LDAL FFFF
7590	02A2	36C1	0000	XR,EOR,AL 0,1
7600	02A3	1201	02A5	JRZ .+2,1
7610	02A4	0000	0000	HLT
7620	02A5	36C1	FFFF	XR,EOR,AL FFFF,1
7630	02A6	1801	02A8	JRO .+2,1
7640	02A7	0000	0000	HLT
7650	02A8	36C1	FFFF	XR,EOR,AL FFFF,1
7660	02A9	1201	02A8	JRZ .+2,1
7670	02AA	0000	0000	HLT
7680	02AB	3A41	FFFF	LDRL FFFF,1
7690	02AC	1801	02AE	JRU .+2,1
7700	02AD	0000	0000	HLT
7710	02AE	36C1	0000	XR,EOR,AL 0,1
7720	02AF	1801	02B1	JRO .+2,1
7730	02B0	0000	0000	HLT
7740	02B1	7A43	0000	LDAL 0
7750	02B2	36C1	FFFF	XR,EOR,AL FFFF,1
7760	02B3	1801	02B5	JRU .+2,1
7770	02B4	0000	0000	HLT
7780	02B5	36C1	0000	XR,EOR,AL 0,1
7790	02B6	1801	02B8	JRO .+2,1
7800	02B7	0000	0000	HLT
7810				* TABLE A2
7820				* CPL
7830	02B8	3041	0000	XR,CK,A 1
7840	02B9	1201	02BB	JRZ .+2,1
7850	02BA	0000	0000	HLT
7860	02BB	3041	0000	XR,CK,A 1
7870	02BC	1801	02BE	JRO .+2,1
7880	02BD	0000	0000	HLT
7890				* NOT A+B
7900	02BE	3141	FFFF	XR,COR,L FFFF,1
7910	02BF	1201	02C1	JRZ .+2,1
7920	02C0	0000	0000	HLT
7930	02C1	3141	FFFF	XR,COR,L FFFF,1
7940	02C2	1201	02C4	JRZ .+2,1
7950	02C3	0000	0000	HLT

7960	02C4	3141	0000	XR,COR,L 0,1
7970	02C5	1801	02C7	JRO .+2,1
7980	02C6	0000	0000	HLT
7990	02C7	3141	0000	XR,COR,L 0,1
8000	02C8	1201	02CA	JRZ .+2,1
8010	02C9	0000	0000	HLT
8020	02CA	3241	FFFF	XR,CKAND,L FFFF,1
8030	02CB	1801	02CD	JRO .+2,1
8040	02CC	0000	0000	HLT
8050	02CD	3241	FFFF	XR,CKAND,L FFFF,1
8060	02CE	1201	02D0	JRZ .+2,1
8070	02CF	0000	0000	HLT
8080	02D0	3241	0000	XR,CKAND,L 0,1
8090	02D1	1201	02D3	JRZ .+2,1
8100	02D2	0000	0000	HLT
8110	02D3	3041	0000	XR,0 1
8120	02D4	1801	02D6	JRO .+2,1
8130	02D5	0000	0000	HLT
8140	02D6	3341	0000	XR,Z 1
8150	02D7	1201	02D9	JRZ .+2,1
8160	02D8	0000	0000	HLT
8170	02D9	3441	FFFF	XR,CAND,L FFFF,1
8180	02DA	1801	02DC	JRO .+2,1
8190	02DB	0000	0000	HLT
8200	02DC	3441	0000	XR,CAND,L 0,1
8210	02DD	1801	02DF	JRO .+2,1
8220	02DE	0000	0000	HLT
8230	02DF	3541	FFFF	XR,CL,L FFFF,1
8240	02E0	1201	02E2	JRZ .+2,1
8250	02E1	0000	0000	HLT
8260	02E2	3541	0000	XR,CL,L 0,1
8270	02E3	1801	02E5	JRO .+2,1
8280	02E4	0000	0000	HLT
8290	02E5	3741	0000	XR,ANDCL,L 0,1
8300	02E6	1801	02E8	JRO .+2,1
8310	02E7	0000	0000	HLT
8320	02E8	3741	FFFF	XR,ANDCL,L FFFF,1
8330	02E9	1201	02EB	JRZ .+2,1
8340	02EA	0000	0000	HLT
8350	02EB	3741	FFFF	XR,ANDCL,L FFFF,1
8360	02EC	1201	02EE	JRZ .+2,1
8370	02ED	0000	0000	HLT
8380	02EE	3841	0000	XR,CKOR,L 0,1
8390	02EF	1801	02F1	JRO .+2,1
8400	02F0	0000	0000	HLT
8410	02F1	3841	0000	XR,CKOR,L 0,1
8420	02F2	1201	02F4	JRZ .+2,1
8430	02F3	0000	0000	HLT
8440	02F4	3841	FFFF	XR,CKOR,L FFFF,1
8450	02F5	1801	02F7	JRO .+2,1
8460	02F6	0000	0000	HLT
8470	02F7	3841	FFFF	XR,CKOR,L FFFF,1
8480	02F8	1801	02FA	JRO .+2,1

8490	02F9	0000	0000	HLT
8500	02FA	3941	0000	XR,EQV,L 0,1
8510	02FB	1201	02FD	JRZ .+2,1
8520	02FC	0000	0000	HLT
8530	02FD	3941	0000	XR,EQV,L 0,1
8540	02FE	1801	0300	JRO .+2,1
8550	02FF	0000	0000	HLT
8560	0300	3A41	0000	XR,LD,L 0,1
8570	0301	1201	0303	JRZ .+2,1
8580	0302	0000	0000	HLT
8590	0303	3A41	FFFF	XR,LD,L FFFF,1
8600	0304	1801	0306	JRO .+2,1
8610	0305	0000	0000	HLT
8620	0306	3841	FFFF	XR,AND,L FFFF,1
8630	0307	1801	0309	JRO .+2,1
8640	0308	0000	0000	HLT
8650	0309	3841	0000	XR,AND,L 0,1
8660	030A	1201	030C	JRZ .+2,1
8670	030B	0000	0000	HLT
8680	030C	3841	FFFF	XR,AND,L FFFF,1
8690	030D	1201	030F	JRZ .+2,1
8700	030F	0000	0000	HLT
8710	030F	3841	0000	XR,AND,L 0,1
8720	0310	1201	0312	JRZ .+2,1
8730	0311	0000	0000	HLT
8740	0312	3D41	FFFF	XR,ORCL,L FFFF,1
8750	0313	1201	0315	JRZ .+2,1
8760	0314	0000	0000	HLT
8770	0315	3D41	0000	XR,ORCL,L 0,1
8780	0316	1801	0318	JRO .+2,1
8790	0317	0000	0000	HLT
8800	0318	3D41	FFFF	XR,ORCL,L FFFF,1
8810	0319	1801	031B	JRO .+2,1
8820	031A	0000	0000	HLT
8830	031B	3D41	0000	XR,ORCL,L 0,1
8840	031C	1801	031E	JRO .+2,1
8850	031D	0000	0000	HLT
8860	031E	3E41	0000	XR,OR,L 0,1
8870	031F	1801	0321	JRO .+2,1
8880	0320	0000	0000	HLT
8890	0321	3E41	FFFF	XR,OR,L FFFF,1
8900	0322	1801	0324	JRO .+2,1
8910	0323	0000	0000	HLT
8920	0324	3901	0001	XR,ADD,L 1,1
8930	0325	3E41	0000	XR,OR,L 0,1
8940	0326	1201	0328	JRZ .+2,1
8950	0327	0000	0000	HLT
8960	0328	3F41	0000	XR,K 1
8970	0329	1201	032B	JRZ .+2,1
8980	032A	0000	0000	HLT
8990	032B	3901	FFFF	XR,ADD,L FFFF,1
9000	032C	1801	032E	JRO .+2,1
9010	032D	0000	0000	HLT

9020	032E	3601	FFFE	XR,SUBM,L -2,1
9030	032F	1201	0331	JRZ .+2,1
9040	0330	0000	0000	HLT
9050	0331	3C01	0000	XR,SL1 1
9060	0332	1201	0334	JRZ .+2,1
9070	0333	0000	0000	HLT
9080	0334	3901	FFFF	XR,ADD,L FFFF,1
9090	0335	3C01	0000	XR,SL1 1
9100	0336	3901	0002	XR,ADD,L 2,1
9110	0337	1201	0339	JRZ .+2,1
9120	0338	0000	0000	HLT
9130	0339	3F01	0000	XR,M1 1
9140	033A	3901	0001	XR,ADD,L 1,1
9150	033B	1201	033D	JRZ .+2,1
9160	033C	0000	0000	HLT
9170	033D	3901	FFFF	XR,ADD,L FFFF,1
9180	033E	3021	0000	XR,P1 1
9190	033F	1201	0341	JRZ .+2,1
9200	0340	0000	0000	HLT
9210	0341	3621	0001	XR,SUB,L 1,1
9220	0342	1801	0344	JRO .+2,1
9230	0343	0000	0000	HLT
9240	0344	3621	FFFF	XR,SUB,L FFFF,1
9250	0345	1201	0347	JRZ .+2,1
9260	0346	0000	0000	HLT
9270	0347	3921	FFFF	XR,ADDP,L FFFF,1
9280	0348	1201	034A	JRZ .+2,1
9290	0349	0000	0000	HLT
9300				' SLIP
9310	034A	3A41	FFFF	LDRL FFFF,1
9320	034B	3C21	FFFF	.INSTRT 36041,FFFF
9330	034C	1801	034E	JRO .+2,1
9340	034D	0000	0000	HLT
9350				' F8
9360	034E	8000	0350	JMP .+2
9370	034F	0000	0000	HLT
9380	0350	8000	0353	JMP .+3
9390	0351	0000	0000	HLT
9400	0352	0000	0000	HLT
9410	0353	8000	0357	JMP .+4
9420	0354	0000	0000	HLT
9430	0355	0000	0000	HLT
9440	0356	0000	0000	HLT
9450	0357	3A41	0002	LDRL 2,1
9460	0358	3A42	0003	LDRL 3,2
9470	0359	8021	0359	JMP .+0,1
9480	035A	0000	0000	HLT
9490	035B	8022	035B	JMP .+0,2
9500	035C	0000	0000	HLT
9510	035D	0000	0000	HLT
9520	035E	8061	035E	J,I .+0,1
9530	035F	0000	0000	HLT
9540	0360	8061	0360	J,I .+0,1

9550	0361	0000	0000	HLT	
9560	0362	0000	0000	HLT	
9570	0363	8041	0363	J,D	+.0,1
9580	0364	0000	0000	HLT	
9590	0365	0000	0000	HLT	
9600	0366	0000	0000	HLT	
9610	0367	8041	0367	J,D	+.0,1
9620	0368	0000	0000	HLT	
9630	0369	0000	0000	HLT	
9640	036A	8080	001A	J	'MCN
9650	036B	0000	0000	HLT	
9660	036C	0000	0000	HLT	
9670	036D			CONS	
9680	036D	8000	0370	J	+.3
9690	036E	0000	0000	HLT	
9700	036F	0000	0000	HLT	
9710	0370	80A2	001B	J	'MCN1,2
9720	0371	0000	0000	HLT	
9730	0372	0000	0000	HLT	
9740	0373	0000	0000	HLT	
9750	0374	80C2	001C	J,D	'MCN2,2
9760	0375	0000	0000	HLT	
9770	0376	0000	0000	HLT	
9780	0377	80C2	001D	J,D	'MCN3,2
9790	0378	0000	0000	HLT	
9800	0379	3902	0001	XR,ADD,L	1,2
9810	037A	80E2	001E	J,I	'MCN4,2
9820	037B	0000	0000	HLT	
9830	037C	0000	0000	HLT	
9840	037D	80E2	001F	J,I	'MCN5,2
9850	037E	0000	0000	HLT	
9860	037F	0000	0000	HLT	
9870	0380	0000	0000	HLT	
9880	0381	0000	0000	HLT	
9890	0382	3622	0014	XR,SUB,L	4,2
9900	0383	1202	0385	JRZ	+.2,2
9910	0384	0000	0000	HLT	
9920	0385	3621	0012	XR,SUB,L	2,1
9930	0386	1201	0388	JRZ	+.2,1
9940	0387	0000	0000	HLT	
9950	0388	8201	038B	JSR	BOO,R1
9960	0389			RBOO\$	
9970	0389	0000	0000	HLT	
9980	038A	0000	0000	HLT	
9990	038B			BOO\$	
10000	038B	3621	0389	XR,SUB,L	RBOO,R1
10010	038C	1201	038E	JRZ	+.2,1
10020	038D	0000	0000	HLT	
10030	038E	3A43	0003	LDRL	3,3
10040				'JSR	INDEXED
10050	038F	8223	0390	.INSTRT	101043,G00
10060	0390			G00\$	
10070	0390	0000	0000	HLT	

10080	0391	0000	0000	HLT
10090	0392	0000	0000	HLT
10100	0393	3623	0390	XR,SUB,L G00,3
10110	0394	1203	0396	JRZ .+2,3
10120	0395	0000	0000	HLT
10130	0396	8281	0020	JSR * H00,1
10140	0397	0000	0000	HLT
10150	0398			LOO%
10160	0398	3621	0397	XR,SUB,L L00-1,1
10170	0399	1201	0398	JRZ .+2,1
10180	039A	0000	0000	HLT
10190	039B	3A44	0004	LDRL 4,4
10200				* JSR *INDEXED
10210	039C	82A4	0021	.INSTRT 101244,FOO
10220	039D	0000	0000	HLT
10230	039E	0000	0000	HLT
10240	039F			MOO%
10250	039F	0000	0000	HLT
10260	03A0	0000	0000	HLT
10270	03A1	0000	0000	HLT
10280	03A2	0000	0000	HLT
10290	03A3	3624	039D	XR,SUB,L MOO-2,4
10300	03A4	1204	03A6	JRZ .+2,4
10310	03A5	0000	0000	HLT
10320	03A6	8A84	0022	.INSTRT 105204,UNO
10330	03A7	1804	03A9	JRO .+2,4
10340	03A8	0000	0000	HLT
10350	03A9	8A84	0023	.INSTRT 105204,ZEO
10360	03AA	1204	03AC	JRZ .+2,4
10370	03AB	0000	0000	HLT
10380				* F CDE
10390	03AC	CA00	0022	LDA UNO
10400	03AD	7903	0001	ADDL 1
10410	03AE	2080	03B0	JEQ .+2
10420	03AF	0000	0000	HLT
10430	03B0	CA00	0023	LDA ZEO
10440	03B1	2080	03B3	JEQ .+2
10450	03B2	0000	0000	HLT
10460				* LDA A-DM
10470	03B3	D600	0022	.INSTRT 153000,UNO
10480	03B4	7623	0001	SUBL 1
10490	03B5	2080	03B7	JEQ .+2
10500	03B6	0000	0000	HLT
10510				* LDA A+DM
10520	03B7	E900	0022	.INSTRT 164400,UNC
10530	03B8	7903	0001	ADDL 1
10540	03B9	2080	03B8	JEQ .+2
10550	03BA	0000	0000	HLT
10560	03BB	7903	0001	ADDL 1
10570	03BC	E900	0022	.INSTRT 164400,UNO
10580	03BD	2080	03BF	JEQ .+2
10590	03BE	0000	0000	HLT
10600	03BF	CA80	0C24	LDA *ARS

10610	03C0	79C3	0001	ADUL 1
10620	03C1	2080	03C3	JEQ .+2
10630	03C2	0000	0000	HLT
10640	03C3	3A41	0007	LDRL 7,1
10650	03C4	CA21	0008	LDA TAB0,1
10660	03C5	7623	0000	SURL 128.
10670	03C6	2080	03C8	JEQ .+2
10680	03C7	0000	0000	HLT
10690	03C8	CA41	0008	LDA,D TAB0,1
10700	03C9	7623	0000	SURL 128.
10710	03CA	2080	03CC	JEQ .+2
10720	03CB	0000	0000	HLT
10730	03CC	6A41	0000	LDAR 1
10740	03CD	7623	0006	SURL 6
10750	03CE	2080	03D0	JEQ .+2
10760	03CF	0000	0000	HLT
10770	03D0	CA61	0008	LDA,I TAB0,1
10780	03D1	7623	0040	SURL 64.
10790	03D2	2080	03D4	JEQ .+2
10800	03D3	0000	0000	HLT
10810	03D4	6A41	0000	LDAR 1
10820	03D5	7623	0007	SUBL 7
10830	03D6	2080	03D8	JEQ .+2
10840	03D7	0000	0000	HLT
10850	03D8	3A41	000A	LDRL 10.,1
10860	03D9	CA41	0025	LDA 'FLOS,1
10870	03DA	7623	0000	SURL 2048.
10880	03DB	2080	03D0	JEQ .+2
10890	03DC	0000	0000	HLT
10900	03DD	3A41	0009	LDRL 9.,1
10910	03DE	CA61	0027	LDA,D 'GLOS,1
10920	03DF	7623	2000	SUBL 8192.
10930	03E0	2080	03E2	JEQ .+2
10940	03E1	0000	0000	HLT
10950	03E2	6A41	0000	LDAR 1
10960	03E3	7623	0008	SUBL 8.
10970	03E4	2080	03E6	JEQ .+2
10980	03E5	0000	0000	HLT
10990	03E6	CA61	0026	LDA,I 'MLOS,1
11000	03E7	7623	0400	SUBL 1024.
11010	03E8	2080	03EA	JEQ .+2
11020	03E9	0000	0000	HLT
11030	03EA	3621	0009	XR,SUB,L 9.,1
11040	03EB	1201	03ED	JRZ .+2,1
11050	03EC	0000	0000	HLT
11060				' AT00M A'TOA
11070	03ED	7A43	0000	LDAL 0
11080	03EE	F000	0029	.INSTRT 170000,FF0
11090	03EF	79C3	0001	ADUL 1
11100	03F0	2080	03F2	JEQ .+2
11110	03F1	0000	0000	HLT
11120	03F2	CA00	0029	LDA FFO
11130	03F3	2080	03F5	JEQ .+2

11140	03F4	0000	0000	HLT
11150	03F5	7A43	FFFF	LDAL FFFF
11160	03F6	F000	0029	.INSTRT 170000,FF0
11170	03F7	2080	03F9	JEQ .+2
11180	03F8	0000	0000	HLT
11190	03F9	CA00	0029	LDA FFO
11200	03FA	7903	0001	ADDL 1
11210	03FB	2080	03FD	JEQ .+2
11220	03FC	0000	0000	HLT
11230				* ATODM 010A
11240	03FD	7A43	AAAA	LDAL AAAA
11250	03FE	F300	0029	.INSTRT 171400,FF0
11260	03FF	2080	0401	JEQ .+2
11270	0400	0000	0000	HLT
11280	0401	CA00	0029	LDA FFO
11290	0402	7623	AAAA	SUBL AAAA
11300	0403	2080	0405	JEQ .+2
11310	0404	0000	0000	HLT
11320	0405	7A43	5555	LDAL 52525
11330	0406	F300	0029	.INSTRT 171400,FF0
11340	0407	2080	0409	JEQ .+2
11350	0408	0000	0000	HLT
11360	0409	CA00	0029	LDA FFO
11370	040A	7623	5555	SUBL 52525
11380	040B	2080	040D	JEQ .+2
11390	040C	0000	0000	HLT
11400	040D	FF00	0029	STA FFO
11410				* ATODM 810A
11420	040E	7A43	FFFF	LDAL FFFF
11430	040F	7903	0000	ADDL 0
11440	0410	F500	0029	.INSTRT 172400,FF0
11450	0411	7903	0001	ADDL 1
11460	0412	2080	0414	JEQ .+2
11470	0413	0000	0000	HLT
11480	0414	CA00	0029	LDA FFO
11490	0415	7903	0001	ADDL 1
11500	0416	2080	0418	JEQ .+2
11510	0417	0000	0000	HLT
11520	0418	7A43	0001	LDAL 1
11530	0419	7903	FFFF	ADDL FFFF
11540	041A	F500	0029	.INSTRT 172400,FF0
11550	041B	2080	041D	JEQ .+2
11560	041C	0000	0000	HLT
11570	041D	CA00	0029	LDA FFO
11580	041E	2080	0420	JEQ .+2
11590	041F	0000	0000	HLT
11600				* ATODM AMOD28TOA
11610	0420	7A43	FFFF	LDAL FFFF
11620	0421	F600	0029	.INSTRT 173000,FF0
11630	0422	2080	0424	JEQ .+2
11640	0423	0000	0000	HLT
11650	0424	CA00	0029	LDA FFO
11660	0425	7903	0001	ADDL 1

11670	0426	2080	0428	JEQ .+2
11680	0427	0000	0000	HLT
11690	0428	7A43	0000	LDAL 0
11700	0429	F600	0029	.INSTRT 173000,FF0
11710	042A	2080	042C	JEQ .+2
11720	042B	0000	0000	HLT
11730	042C	CA00	0029	LDA FF0
11740	042D	2080	042F	JEQ .+2
11750	042E	0000	0000	HLT
11760	042F	7A43	0001	LDAL 1
11770	0430	7903	FFFF	ADDL FFFF
11780	0431	F600	0029	.INSTRT 173000,FF0
11790	0432	7903	0001	ADDL 1
11800	0433	2080	0435	JEQ .+2
11810	0434	0000	0000	HLT
11820	0435	CA00	0029	LDA FF0
11830	0436	2080	0438	JEQ .+2
11840	0437	0000	0000	HLT
11850	0438	7A43	FFFF	LDAL FFFF
11860	0439	7903	0000	ADDL 0
11870	043A	F600	0029	.INSTRT 173000,FF0
11880	043B	7903	0001	ADDL 1
11890	043C	2080	043E	JEQ .+2
11900	043D	0000	0000	HLT
11910	043E	CA00	0029	LDA FF0
11920	043F	7903	0001	ADDL 1
11930	0440	2080	0442	JEQ .+2
11940	0441	0000	0000	HLT
11950				* ATODM BTOA
11960	0442	7A43	0001	LDAL 1
11970	0443	7903	FFFF	ADDL FFFF
11980	0444	FA00	0029	.INSTRT 175000,FF0
11990	0445	7903	0001	ADDL 1
12000	0446	2080	0448	JEQ .+2
12010	0447	0000	0000	HLT
12020	0448	CA00	0029	LDA FF0
12030	0449	2080	044B	JEQ .+2
12040	044A	0000	0000	HLT
12050	044B	7A43	FFFF	LDAL FFFF
12060	044C	7903	0000	ADDL 0
12070	044D	FA00	0029	.INSTRT 175000,FF0
12080	044E	2080	0450	JEQ .+2
12090	044F	0000	0000	HLT
12100	0450	CA00	0029	LDA FF0
12110	0451	7903	0001	ADDL 1
12120	0452	2080	0454	JEQ .+2
12130	0453	0000	0000	HLT
12140				* ATODM ABTOA
12150	0454	7A43	0000	LDAL 0
12160	0455	F600	0029	.INSTRT 175400,FF0
12170	0456	2080	0458	JEQ .+2
12180	0457	0000	0000	HLT
12190	0458	CA00	0029	LDA FF0

12200	0459	2080	0458	JEQ .+2
12210	045A	0000	0000	HLT
12220	045B	7A43	FFFF	LDAL FFFF
12230	045C	7903	0000	ADDL 0
12240	045D	FB00	0029	.INSTRT 175400,FF0
12250	045E	2080	0460	JEQ .+2
12260	045F	0000	0000	HLT
12270	0460	CA00	0029	LDA FF0
12280	0461	7903	0001	ADDL 1
12290	0462	2080	0464	JEQ .+2
12300	0463	0000	0000	HLT
12310	0464	7A43	0001	LDAL 1
12320	0465	7903	FFFF	ADDL FFFF
12330	0466	FB00	0029	.INSTRT 175400,FF0
12340	0467	2080	0469	JEQ .+2
12350	0468	0000	0000	HLT
12360	0469	CA00	0029	LDA FF0
12370	046A	2080	046C	JEQ .+2
12380	046B	0000	0000	HLT
12390	046C	7A43	FFFF	LDAL FFFF
12400	046D	FB00	0029	.INSTRT 175400,FF0
12410	046E	7903	0001	ADDL 1
12420	046F	2080	0471	JEQ .+2
12430	0470	0000	0000	HLT
12440	0471	CA00	0029	LDA FF0
12450	0472	7903	0001	ADDL 1
12460	0473	2080	0475	JEQ .+2
12470	0474	0000	0000	HLT
12480				* ATODM 1TOA
12490	0475	7A43	0000	LDAL 0
12500	0476	FB00	0029	.INSTRT 176000,FF0
12510	0477	7903	0001	ADDL 1
12520	0478	2080	047A	JEQ .+2
12530	0479	0000	0000	HLT
12540	047A	CA00	0029	LDA FF0
12550	047B	2080	047D	JEQ .+2
12560	047C	0000	0000	HLT
12570	047D	7A43	FFFF	LDAL FFFF
12580	047E	FB00	0029	.INSTRT 176000,FF0
12590	047F	7903	0001	ADDL 1
12600	0480	2080	0482	JEQ .+2
12610	0481	0000	0000	HLT
12620				* ATODM A+8TOA
12630	0482	7A43	0001	LDAL 1
12640	0483	7903	FFFF	ADDL FFFF
12650	0484	FB00	0029	.INSTRT 177000,FF0
12660	0485	7903	0001	ADDL 1
12670	0486	2080	0488	JEQ .+2
12680	0487	0000	0000	HLT
12690	0488	CA00	0029	LDA FF0
12700	0489	2080	048B	JEQ .+2
12710	048A	0000	0000	HLT
12720	048B	7A43	FFFF	LDAL FFFF

12730	048C	7903	0000	ADDL 0
12740	048D	FF00	0029	.INSTRT 177000,FF0
12750	048F	7903	0001	ADDL 1
12760	048F	2080	0491	JEQ .+2
12770	0490	0000	0000	HLT
12780	0491	CA00	0029	LDA FF0
12790	0492	7903	0001	ADDL 1
12800	0493	2080	0495	JEQ .+2
12810	0494	0000	0000	HLT
12820	0495	7A43	E1E1	LDAL 160741
12830	0496	7903	0F0F	ADDL 7417
12840	0497	FF00	0029	.INSTRT 177000,FF0
12850	0498	7903	0001	ADDL 1
12860	0499	2080	0498	JEQ .+2
12870	049A	0000	0000	HLT
12880	049B	CA00	0029	LDA FF0
12890	049C	7623	FCFC	SUBL 170360
12900	049D	2080	049F	JEQ .+2
12910	049E	0000	0000	HLT
12920	049F	CA00	000B	LDA TAB0
12930	04A0	FF00	000B	STA TAB0
12940	04A1	D600	000B	SUB TAB0
12950	04A2	2080	04A4	JEQ .+2
12960	04A3	0000	0000	HLT
12970	04A4	7A43	0000	LDAL 0
12980	04A5	FF00	002E	STA FRD
12990	04A6	7A43	FFFF	LDAL FFFF
13000	04A7	FF80	002B	STA 'CST0
13010	04A8	CA00	002E	LDA FRD
13020	04A9	7903	0001	ADDL 1
13030	04AA	2080	04AC	JEQ .+2
13040	04AB	0000	0000	HLT
13050	04AC	FF00	002E	STA FRD
13060	04AD	3A41	0002	LDR 2,1
13070	04AE	7A43	FFFF	LDAL FFFF
13080	04AF	FF21	002F	STA FRD,1
13090	04B0	CA00	0030	LDA LUC
13100	04B1	7903	0001	ADDL 1
13110	04B2	2080	04B4	JEQ .+2
13120	04B3	0000	0000	HLT
13130	04B4	FF00	0030	STA LUC
13140	04B5	3A42	0003	LDRL 3,2
13150	04B6	7A43	FFFF	LDAL FFFF
13160	04B7	FF62	002F	STA,I BUC,2
13170	04B8	7903	0001	ADDL 1
13180	04B9	CA00	0032	LDA MOC
13190	04BA	7903	0001	ADDL 1
13200	04BB	2080	04BD	JEQ .+2
13210	04BC	0000	0000	HLT
13220	04BD	FF00	0037	STA MOC
13230	04BE	3622	0004	XR,SUB,L 4,2
13240	04BF	1207	04C1	JRZ .+2,2
13250	04C0	0000	0000	HLT

13260	04C1	39C1	0001	XR,ADD,L 1,1
13270	04C2	7A43	FFFF	LDAL FFFF
13280	04C3	FF41	002E	STA ,D FRD,1
13290	04C4	79C3	0001	ADDL 1
13300	04C5	CAC0	0031	LDA HOC
13310	04C6	7903	0001	ADDL 1
13320	04C7	2C80	04C9	JEQ .+2
13330	04C8	0000	0000	HLT
13340	04C9	3621	0002	XR,SUB,L 2,1
13350	04CA	1201	04CC	JRZ .+2,1
13360	04CB	0000	0000	HLT
13370	04CC	FF00	0031	STA HOC
13380	04CD	3A41	0003	LDRL 3,1
13390	04CE	7A43	FFFF	LDAL FFFF
13400	04CF	FFA1	002C	STA 'CSTT,1
13410	04D0	79C3	0001	ADDL 1
13420	04D1	CAC0	0036	LDA CVY+3
13430	04D2	7903	0001	ADDL 1
13440	04D3	2C80	04D5	JEQ .+2
13450	04D4	0000	0000	HLT
13460	04D5	FF00	0036	STA CVY+3
13470	04D6	7A43	FFFF	LDAL FFFF
13480	04D7	FFC1	002D	STA,D 'CSTF,1
13490	04D8	79C3	0001	ADDL 1
13500	04D9	CAC0	004C	LDA DGE+3
13510	04DA	7903	0001	ADDL 1
13520	04DB	2C80	04DD	JEQ .+2
13530	04DC	0000	0000	HLT
13540	04DD	FF00	004D	STA DGE+3
13550	04DE	3621	0002	XR,SUB,L 2,1
13560	04DF	1201	04E1	JRZ .+2,1
13570	04E0	0000	0000	HLT
13580	04E1	3A41	0005	LDRL 5,1
13590	04E2	7A43	FFFF	LDAL FFFF
13600	04E3	FFE1	002D	STA,I 'CSTF,1
13610	04E4	79C3	0001	ADDL 1
13620	04E5	CAC0	0042	LDA DGE+5
13630	04E6	7903	0001	ADDL 1
13640	04E7	2C80	04E9	JEQ .+2
13650	04E8	0000	0000	HLT
13660	04E9	FF00	0042	STA DGE+5
13670	04EA	3621	0006	XR,SUB,L 6,1
13680	04EB	1201	04ED	JRZ .+2,1
13690	04EC	0000	0000	HLT
13700	04ED	7A43	7FFF	LDAL 77777
13710	04EE	7903	0001	ADDL 1
13720	04EF	7903	7FFF	ADDL 77777
13730	04F0	7903	0001	ADDL 1
13740	04F1	3A40	04F5	XR,LD,L .+4,0
13750	04F2	8420	0000	JRI 0,0
13760	04F3	0000	0000	HLT
13770	04F4	0000	0000	HLT
13780	04F5	08E0	0000	DINT

13790	04F6	7903	8000	ADDL 100000	
13800	04F7	7903	8000	ADDL 100000	
13810	04F8	2080	04FA	JEQ .+2	
13820	04F9	0000	0000	HLT	
13830	04FA	3A40	04FD	XR,LD,L .+3,0	
13840	04FB	8420	0000	JRI 0,0	
13850	04FC	0000	0000	HLT	
13860	04FD	0800	0000	DINT	
13870	04FE	2080	0500	JEQ .+2	
13880	04FF	0000	0000	HLT	
13890	0500	CA00	0047	LDA LOGC	jmp 502
13900	0501	2020	0508	JBR 0,SECD	
13910	0502	0CE0	0000	EINT	
13920	0503	8000	0503	JMP .+0	
13930	0504			WXYZ\$	
13940	0504	3620	0503	XR,SUB,L .-1,0	
13950	0505	0040	0000	INPS 0	
13960	0506	1200	050F	JRZ BGN,0	
13970	0507	0000	0000	HLT	
13980	0508			SECD\$	
13990	0508	0CE0	0000	EINT	
14000	0509	8000	0509	JMP .+0	
14010	050A			ZYXW\$	
14020	050A	3620	0509	XR,SUB,L .-1,0	
14030	050B	0F40	0000	INPS 17	
14040	050C	1200	050F	JRZ BGN,0	
14050	050D	0000	0000	HLT	
14060	050E	0000	0000	HLT	
14070	050F			BGN\$	
14080	050F	CA00	0047	LDA LOGC	
14090	0510	7903	0001	ADDL 1	
14100	0511	FF00	0047	STA LOGC	
14110	0512	2000	0518	JRS 0,ODD	
14120	0513	7A43	FFFF	LDAL FFFF	
14130	0514	0F60	0000	OUTA 17	
14140	0515	7A43	0000	LDAL 0	
14150	0516	0E60	0000	OUTA 16	
14160	0517	8000	051C	JMP .+5	
14170	0518			ODD\$	
14180	0518	7A43	0000	LDAL 0	
14190	0519	0F60	0000	OUTA 17	
14200	051A	7A43	FFFF	LDAL FFFF	
14210	051B	0E60	0000	OUTA 16	
14220	051C	7A43	0000	LDAL 0	
14230	051D	FF00	0000	STA RM1	
14240	051E	FF00	0001	STA RM2	
14250	051F	3AC1	FFFF	STAR 1	
14260	0520	3AC2	FFFF	STAR 2	
14270	0521	3AC3	FFFF	STAR 3	
14280	0522	7A43	FFFF	LDAL FFFF	
14290	0523	3AC4	FFFF	STAR 4	
14300	0524	8000	0010	JMP GO1	
14310	0525	4F43	0000	NUP	

14320	0526	8420	0000	JRI C,0
14330	0527	0000	0000	EINT
14340	0528	8000	0528	JMP .+0
14350	0529	0000	0000	HLT
14360	052A	0000	0000	HLT
14370	052B			AOUT\$
14380	052B	0000	0000	HLT
14390	052C			ROUT\$
14400	052C	0000	0000	HLT
14410	052D			COUT\$
14420	052D	0000	0000	HLT
14430	052F			DOUT\$
14440	052E	0000	0000	HLT
14450	052F			EOUT\$
14460	052F	0000	0000	HLT
14470	0530			FOUT\$
14480	0530	0000	0000	HLT
14490	0531			GOUT\$
14500	0531	0000	0000	HLT
14510	0532			HOUT\$
14520	0532	0000	0000	HLT
14530	0533			IOUT\$
14540	0533	0000	0000	HLT
14550	0534			JOUT\$
14560	0534	0000	0000	HLT
14570	0535			KOUT\$
14580	0535	0000	0000	HLT
14590	0536			LOUT\$
14600	0536	0000	0000	HLT
14610	0537			MOUT\$
14620	0537	0000	0000	HLT
14630	0538			NOUT\$
14640	0538	0000	0000	HLT
14650	0539			OOOUT\$
14660	0539	0000	0000	HLT
14670	053A			POUT\$
14680	053A	0000	0000	HLT
14690	053B			OUT1\$
14700	053B	0000	0000	HLT
14710	053C			OUT2\$
14720	053C	0000	0000	HLT
14730	053D	0000	0000	HLT
14740	053E	0000	0000	HLT
14750	053F	0000	0000	HLT
14760	0540			OUTA\$
14770	0540	0000	0000	HLT
14780	0541			OUTB\$
14790	0541	0000	0000	HLT
14800	0542			OUTC\$
14810	0542	0000	0000	HLT
14820	0543			OUTD\$
14830	0543	0000	0000	HLT
14840	0544			OUTE\$

AD-A074 321

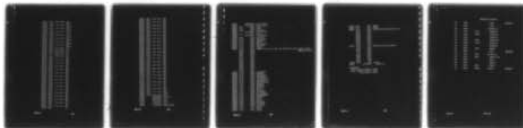
GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 9/2  
HF PROGRAMMABLE MODEM. (U)  
OCT 74

N00014-74-C-0299  
NL

UNCLASSIFIED

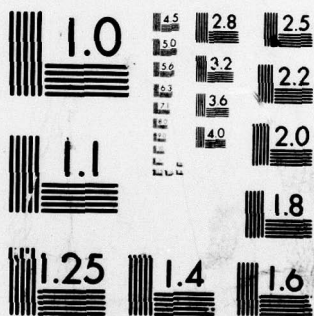
3 OF 3

AD  
A074321



END  
DATE  
FILMED

10-19  
DDC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

14850	0544	0000	0000	HLT
14860	0545			OUTF\$
14870	0545	0000	0000	HLT
14880	0546			OUTG\$
14890	0546	0000	0000	HLT
14900	0547			OUTH\$
14910	0547	0000	0000	HLT
14920	0548			OUTI\$
14930	0548	0000	0000	HLT
14940	0549			OUTJ\$
14950	0549	0000	0000	HLT
14960	054A			OUTK\$
14970	054A	0000	0000	HLT
14980	054B			OUTL\$
14990	054B	0000	0000	HLT
15000	054C			OUTM\$
15010	054C	0000	0000	HLT
15020	054D	0000	0000	HLT
15030	054E	0000	0000	HLT
15040	054F	0000	0000	HLT
15050	0550	0000	0000	HLT
15060	0551	0000	0000	HLT
15070	0552			OAS
15080	0552	0000	0000	HLT
15090	0553			OB\$
15100	0553	0000	0000	HLT
15110	0554			OC\$
15120	0554	0000	0000	HLT
15130	0555			OD\$
15140	0555	0000	0000	HLT
15150	0556			OE\$
15160	0556	0000	0000	HLT
15170	0557			OF\$
15180	0557	0000	0000	HLT
15190	0558			OG\$
15200	0558	0000	0000	HLT
15210	0559			AX\$
15220	0559	0000	0000	HLT
15230	055A			BX\$
15240	055A	0000	0000	HLT
15250	055B			CX\$
15260	055B	0000	0000	HLT
15270	055C			DX\$
15280	055C	0000	0000	HLT
15290	055D			ZA\$
15300	055D	0000	0000	HLT
15310	055E			ZB\$
15320	055E	0000	0000	HLT
15330	055F			ZC\$
15340	055F	0000	0000	HLT
15350	0560			ZD\$
15360	0560	0000	0000	HLT
15370	0561			ZE\$

15380	0561	0000	0000	HLT
15390	0562			ZF\$
15400	0562	0000	0000	HLT
15410	0563			ZG\$
15420	0563	0000	0000	HLT
15430	0564			ZH\$
15440	0564	0000	0000	HLT
15450	0565			ZI\$
15460	0565	0000	0000	HLT
15470	0566			ZJ\$
15480	0566	0000	0000	HLT
15490	0567			ZK\$
15500	0567	0000	0000	HLT
15510	0568			ZL\$
15520	0568	0000	0000	HLT
15530	0569			ZM\$
15540	0569	0000	0000	HLT
15550	056A			ZN\$
15560	056A	0000	0000	HLT
15570	056B			ZO\$
15580	056B	0000	0000	HLT
15590	056C			ZP\$
15600	056C	0000	0000	HLT
15610	056D			ZQ\$
15620	056D	0000	0000	HLT
15630	056E			ZR\$
15640	056E	0000	0000	HLT
15650	056F			ZS\$
15660	056F	0000	0000	HLT
15670	0570			ZT\$
15680	0570	0000	0000	HLT
15690	0571			ZU\$
15700	0571	0000	0000	HLT
15710	0572			ZV\$
15720	0572	0000	0000	HLT
15730	0573			ZW\$
15740	0573	0000	0000	HLT
15750	0574			ZX\$
15760	0574	0000	0000	HLT
15770	0575			ZY\$
15780	0575	0000	0000	HLT
15790	0576			ZZ\$
15800	0576	0000	0000	HLT
15810	0577			ZAA\$
15820	0577	0000	0000	HLT
15830	0578			ZAB\$
15840	0578	0000	0000	HLT
15850		000000000		.DATA 0
15860		000000001		R1=1
15870		000000002		R2=2
15880		000000003		R3=3
15890		000000004		R4=4
15900		00000AAAA		AAAA=125252

15910		00000AAA8	AAAR=AAAA+1
15920		00000AAA9	AAA9=125251
15930	0000	0001	RM1\$1
15940	0001	0001	RM2\$1
15950	0002	FFFF	FULL\$FFFF
15960		00000FFFF	FFFF=177777
15970	0003	0000	ENTY\$0
15980		00000FFFE	FFFE=FFFF-1
15990	0004	FFFF	FRST\$FFFF
16000	0005	5555	SND\$52525
16010	0006	AAAA	MEM\$AAAA
16020	0007	0000	ZERO\$0
16030	0008	5555	OMEM\$52525
16040	0009	AAAA	TRE\$AAAA
16050	000A	0000	MRK2\$0
16060	000B	0001	TAB0\$1,2,4,8.,16.,32.,64.,128.,512.,1024.,2048.,4096.,
	000C	0002	8192.,16384.
	000D	0004	
	000F	0008	
	000F	0010	
	0010	0020	
	0011	0040	
	0012	0080	
	0013	0200	
	0014	0400	
	0015	0800	
	0016	1000	
	0017	2000	
	0018	4000	
16070	0019	000B	1TAB\$TAB0
16080	001A	0360	MCN\$CON
16090	001B	0371	MCN1\$CON+4
16100	001C	0374	MCN2\$CON+7
16110	001D	0377	MCN3\$CON+10.
16120	001E	037B	MCN4\$CON+14.
16130	001F	037F	MCN5\$CON+18.
16140	0020	0398	H00\$LOU
16150	0021	039F	F00\$M00
16160	0022	FFFF	UN0\$FFFF
16170	0023	0000	ZEN\$0
16180	0024	0022	ARS\$UNU
16190	0025	000B	FLN\$TAB0
16200	0026	000C	HL0\$TAB0+1.
16210	0027	000E	GLO\$TAB0+3
16220	0028	FFFF	FF\$FFFF
16230	0029	0000	FF0\$0
16240	002A	0000	FFT\$0
16250	002B	002E	CST0\$FRD
16260	002C	0033	CSTT\$CVY
16270	002D	003D	CSTF\$DGE
16280	002E	000C	FRD\$0
16290	002F	0000	HUC\$0
16300	0030	0000	LUC\$0

16310	0031	0000	HOC\$0
16320	0032	0000	MOC\$0
16330	0033	FFFF	CVY\$FFFF,0,0,0,0,C,C,0,0, 0
	0034	0000	
	0035	0000	
	0036	0000	
	0037	0000	
	0038	0000	
	0039	0000	
	003A	0000	
	003B	0000	
	003C	0000	
16340	003D	0000	DGE\$0,0,0,C,0,0,C,0,0,0
	003E	0000	
	003F	0000	
	0040	0000	
	0041	0000	
	0042	0000	
	0043	0000	
	0044	0000	
	0045	0000	
	0046	0000	
16350	0047	0001	LOGC\$1
16360		000000004	.END STRT

CHECKSUMS

LEFT PROGRAM MEMORY= 0AB13  
 RIGHT PROGRAM MEMORY= 0FB78  
 DATA MEMORY= 09BE5

# PROGRAM LISTING

00	3340		CLR	(START)
01	3341		CLR	
02	6A40		A, LDAR R0	
03	9421	0000	MLNR R1	
04	3AC6	FFFF	STAR R6	
05	6A41		LDAR R1	
06	9420	0000	MLNR R0	
07	3AC7	FFFF	STAR R7	
08	6626		SUBR R6	
09	2080	000B	JEQ B	
0A	0000		HLT	(ERROR)
0B	1801	0010	B, JRO C, 1	
0C	1460	0002	JRNO, I A, 2	
0D	3340		CLR	
0E	3021		INR R1	
0F	8000	0002	JMP A	
10	0000	FFFF	C, HLT	(FINISH)